

S I N C L A I R

Every month £1.75 AUGUST 1990

4 10 7 9
13 14 15
24 26 34

QL

WORLD

MULTIBASIC

Revised, expanded and concluded

Software File:

**Pro Pub
Deluxe
Font
Enlarger**

QPAC 2 – a great new
update to the original
utility

FLEET TACTICAL COMMAND

Reviewed by Bryan Davies
From the Authors – the Whole Story

Software file:

Uncle Looney's Legacy
The Progs: Cipher

ARCHIVE POWER PART-3

ISSN 0951-9335



9 770951 933009

08

SINCLAIR



Editor
Helen Armstrong

Production Controller
Jayne Penfold

Designer
Jeff Gurney

Advertising Sales
Jason Newman

Magazine Services
Sheila Baker

Advertising Production
Michelle Evans

Group Advertising Manager
Richard Vaughan

Group Editor
John Taylor

Publishing Director
Ray Lewis

Managing Director
Peter Welham

Sinclair QL World
Panini House
116-120 Goswell Road
London EC1V 7QD
Telephone 071-490 7161
ISSN 026806X

Unfortunately, we are no longer able to answer enquiries made by telephone. If you have any comments or difficulties, please write to The Editor, Open Channel, Trouble Shooter, or Psion Solutions. We will do our best to deal with your problem in the magazine, though we cannot guarantee individual replies. Back issues are available from the publisher price £2 U.K., £2.75 Europe. Overseas rates on request. Published by Maxwell Consumer Magazines, A Division of MCPC Ltd. S M Distribution, Streatham, London SW1. 01 677 8111. Subscription information from: MCM Subscription Dept, Lazahold Ltd., PO Box 10, Roper St, Pallion Ind. Est., Sunderland SR4 4SN. £21.00 U.K., £24.70 Europe, Middle East £25.80, Far East £27.60, Rest of World £26.20, U.S.A. \$45.00. Airmail rates available on request. Typesetting by Adtec Typographics, Britannia Court, Basildon, Essex. Tel: (0268) 591110. Printing by Cradley Print. Sinclair QL World is published on the fourth Wednesday preceding cover date.
© COPYRIGHT
SINCLAIR QL WORLD — 1990.

CONTENTS

■ ■ AUGUST 1990

- 9 QL SCENE ● Athene on Turboquill+
- 10 OPEN CHANNEL ● Disks and microdrives
- 12 SOFTWARE FILE ● Uncle Looney's Legacy
- 13 QL SCENE ● Ablex will take orders
- 14 TROUBLESHOOTER ● New Spellbound
- 18 FLEET TACTICAL COMMAND ● The True Story
- 20 SOFTWARE FILE ● Fleet Tactical Command
- 24 SOFTWARE FILE ● Deluxe Font Enlarger
- 26 DIY TOOLKIT ● Multibasic concluded and updated
- 33 NEW COVER ● A preview of our new cover design
- 34 ARCHIVE POWER ● Part three of a four-part series
- 39 SOFTWARE FILE ● QPAC 2 — the update
- 42 SUBSCRIPTION INFORMATION
- 44 THE PROGS ● Ciphers
- 49 MICRODRIVE EXCHANGE ● Gradual progression



NEXT MONTH DIY TOOLKIT

Some new routines to pull together earlier DIY Toolkits towards a unified system.

NEW COVER

QL World has a new full-colour cover design from next month. See a mono preview on page 33.

QL

SCENE

Ablex can take microcassette orders

Ablex, manufacturers of the Sinclair microcassettes, are not yet in full production of the revived cartridges, but are able to supply small and medium-sized orders

"Ideally we would like orders for about 1,000", said Production Manager David MacSorley, speaking to *QL World* in mid-June, "But we can supply small orders if people phone us for a quotation first, and send

cash with order." The number for enquiries is 0952 680131, and ask to speak to David MacSorley.

Ablex name ICL and British Telecom, both of which use the ICL One Per Desk internally, as continuing to buy microcassettes in some quantity, but a former steady customer, the Boots chemist chain, had cancelled its orders, MacSorley said.

This will come as a disappointment to those users who looked to larger Boots branches as their main source of supply.

A stock of mdvs is also held by EEC, who bought up Ablex's running stock 'at quite a high price' when the impending shortage was announced, as a support for the stock of QL microcassette drive units which they hold. *QL World* spoke to

EEC this week, and Bill Richardson confirmed that they have "plenty of QLs" and parts in stock.

TK Comware and Sector Software are also established customers who are still placing regular orders. David Batty has informed us that the microdrive Exchange will now be able to supply microcassettes at £2 a piece.

MS-QLink – the Revisions

Our statement at the end of the review of *MS-QLink* (*QL World* June 1990) that a list of subsequent revisions had appeared in *QL Scene*, *QL World* May 1990, was in error. The list of upgrades which appeared in the May issue related to Text87.

The list of revisions which appear in MS-QLink V1.3 are as follows:

The boot procedure now works without hitch on the JM QL with a Trump Card, with and without TK2 activated. In case of trouble, a short routine to make new boot files is provided, mainly for AH and JM QL users. The problems previously noted when changing default drive, and when asking for directories, did not show themselves with the new version. If a requested drive is not available, or a disk in a drive is of the wrong type, an appropriate message is displayed.

MS-DOS sub-directories are now supported in so far as they are listed, but the files in them cannot be accessed (for deletion, etc). Taking a directory of an MS-DOS disk with sub-directories on it, and with files in the root directory, now gives the full information, all root files being listed with their correct lengths. The volume name is displayed, and all files

can be erased, now as expected.

There is no problem with "missing" MS-DOS or QDOS files. Formatting a QDOS disc in both flp1 and flp2 gives 1,440 sectors. The unwanted space previously added to the extension of file names during the MS-DOS-to-QDOS conversion process is no longer.

A point not made when the review was written is that the XOver transfer utility also comes with *Media Manager Special Edition*.

In summary, the reservations expressed in the review do not apply to version 1.3. It can be recommended as a cheap and useful utility program

which is fully usable. Further developments are anticipated, but there is now no reason to wait until they appear before purchasing.

Discopy has now reached version 2.0, and the user interface is simple and effective. The author hopes to have this ready for sale shortly.

The programs cannot at present be supplied for microdrives "because there are no longer any cartridges in Portugal"! MDV users might like to discuss the possibility of supplying their own cartridges with the publisher.

MS-QLink costs £12 from Qfile, Apartado 2110, 1103 Lisboa Codex, Portugal.



Ant's Move

Ant Publishing, publishers of *Chinese Chess*, have moved from their previous site in Huddersfield to an address in Oxfordshire. They can now be contacted at 11 Latton Close, Chilton, Didcot, Oxon OX11 0SU. Tel: 0235 834254.

Chinese Chess is currently available price at £14.95. A software list is available from Ant, who also publish for other machines.

FLEET CLUB

Users of *Fleet Tactical Command*, reviewed in this issue on page 20, now have their own user group providing help and materials for the game. The news letter, FT.ComClub, will appear periodically. New charts and logbooks for the game will be available, plus

news of new releases from the publishers, Di-ren, plus upgrade news, special offers and maybe most important, a user-contact service to put players in contact with other users upon request.

Hints on both game-play and on networking and using the QL effectively are included.

Contact the publishers at **FT.ComClub, 43 Davids Road, Forest Hill, London SE23 3EP** if you want more information:

The accredited FTC dealers for West Germany are now **QLympic CS**, who will be setting up a User Group/helpline for players in Germany.

OPEN CHANNEL

Open Channel is where you have the opportunity to voice your opinions in *Sinclair QL World*. Whether you want to ask for help with a technical problem, provide somebody

with the answer, or just sound off about something which bothers you, write to: Open Channel, Sinclair QL World, 116/120 Goswell Road, London EC1V 7QD.

Disks

I am one of those owners who realised the non-availability of microdrives meant that I would have to go on to disk drive. The cost was a deterrent but after attending the Northern Micro-fair last year I realised that the price need not be as high as I thought. After a chat to various exhibitors I bought a bare drive, a power supply and an interface. I also received from Adman Services all the advice I needed plus a "helpline" phone number, which I have found invaluable.

I met various snags, but none that Adman didn't help me to sort out, and have now added a second drive unit. It all works! The cost for interface (with toolkit) and twin drive has

been around £130 plus my time and half a dozen phone calls. May I, through your pages, pay tribute to the helpfulness of Dennis Briggs at Adman, and encourage anyone else wanting to save money to visit a micro-fair, ask for advice and then to have a go.

Martin Baxter,
Ulverston,
Cumbria.

Blue!

Having just fitted a new 512K ram and loaded Psion's *Quill*. I banged in a few words, then hit the <ENTER> button. The instant I did so, the whole house and my work station lurched and rattled. This completely unexpected and disturbing effect lasted 3-4

seconds, but I did notice the time; it was a minute or so past a quarter to three, Monday 2nd April. I had no idea that my modestly up-graded QL would be so powerful! Future up-graders be warned.

With the same peeking-out-of-the-end-ram I decided to run a little SBasic routine which I knew probably would not work on an extended machine - and it didn't! I powered down, removed the ram, powered up, pressed F1 for Mode 4, and the monitor screen (RGB 12in Vision QL) did its usual thing and remade itself - but in black and blue!

Consternation! What have I done? Blown a chip? How much for repair? When the black (and blue) mists of despair receded and rational thought processes again took charge, I eventually found that the RGB plug on the back of the QL was not firmly in place. Pushed it home and voila! Normal red, green, white and black. But can anyone out there tell me how to invoke and use the blue in a program, in Mode 4? After all, it was a very nice electric blue.

By the way, how does one pronounce Psion - Pi'son; P'sion; or with a silent (p) sion, as in psycho?

David Barnes
Taunton
Somerset

Editors' comment: With a silent P as in (p)sycho of course. Fancy missing a clue like that!

Help

Please can anyone offer me assistance with *Assembler Workbench*? I should have been warned by your article on Monitors in QLW September 1986. The documentation on the cartridge is absolutely no use to one such as myself with only a rudimentary knowledge of programming. I did think

that the lapse of three years might have brought an improvement in the documentation, but it is absolutely unintelligible to me.

I would appreciate suggestions of any available reading matter that may help me to get this program running.

E. F. King
8 Winds Ridge,
Send Hill,
Send, Woking,
Surrey GU23 7HU

No Graphics

I am very desperate and at my wits end with a problem I have with printing graphics on my printer. I can't. Please can anyone out there help? I have a Centronics GPL II dot matrix printer which I cannot get to print graphics. I have tried all sorts of things but nothing seems to work. The only thing that happens when I ask for a printout is a few undistinguishable lines and then the paper moves up a complete sheet and continues repeating this. I have three pieces of software (*Easel*, *Eye Q* and *Professional Publisher*) all of which use graphics to print and all do the same as described above. Text printing, by the way, from *Quill*, *Abacus* or *Archive* generally works well.

I use the Miracle Systems QL Centronics interface for printing which is working OK as I was able to borrow an Epson LX80 which printed the graphics from my three pieces of software. So if there is anyone out there who has any sympathy for a man with no graphics please contact me.

S Curtis,
3 Avenue House,
Cazenove Road,
Stoke Newington,
London N16 6AP

Editors Comment: Have you spoken to Miracle Systems?

Editor's notebook

Firstly, watch out - next month - for the new *QL World* cover design. We thought that it was time to celebrate our arrival in a new home, in a new decade, with a new up-to-date colour cover. The 'traditional' *QL World* logo has been preserved, but in a very different form. There's a black-and-white preview on page 33 - tell us what you think of it.

Secondly, the Microdrive Exchange is progressing more slowly than we had hoped. Weeks fly by, one or two authors are proving difficult to locate, and just this morning we received an envelope with a neat little holed chewed in one end where another author's mdv had made a successful break for freedom. Tape them down! You wouldn't ride on the Astro Whirly Wheel without a seat-belt, would you? Be gentle with mdvs.

Apart from that, things are progressing well and the MDX can now supply microcassettes again, as well as having a few more returnees to the list.

The office staff of *QL World* regrets that it has been unable to locate Mr. John Silk for comment on the news story which appears in *QL Scene* this month, although we would be pleased to hear from anybody who has spoken to him recently.

Bulk

I have a JM 'Rom QL, expanded to 896K, using a Trump Card and twin floppies.

I have often thought how nice it would be to be able to use all those wonderful programs by Simon Goodwin and others that appear in QL World but are not incorporated in the "Microdrive Exchange". I lack the energy, and am too busy to type them in myself.

Since they must have been typed originally by the authors, it is a waste of time to have to do it again. Why not issue a 3.5in or 5.25in disk, with the appropriate limited life, of all programs appearing in each month's magazine? I am sure this would generate income for you and you would be able to buy the disks in bulk. It would not matter if disks were issued a month later than the magazine in which they were printed.

**Arthur Cartwright,
90 Harlands Road,
Haywards Heath,
RH16 1LS.**

Editors Comment: We haven't been in the right circumstances until recently, but this is something we will be looking at when the MDX is back off the ground.

Mcorrection

I have two small corrections for my article Abacus Spreadsheet (QL World, July 1990). The first is: the Mc is missing from my name in the byline. The second is one of my own: int command for cell L21 has been omitted, and should read:

CELL L21 F3 E J21 over
range M21:V21

**James McGreehin
Alva
Scotland**

Ireland

In spite of rumours to the contrary, Quanta is alive and well in Northern Ireland.

To date we have had two successful meetings, both well attended. Fourteen out of the twenty Quanta members in

Northern Ireland have shown positive interest.

We meet the last Saturday of each month at 2.30pm, except when it clashes with a Bank Holiday weekend.

Venues vary, so please phone 06937 38567.

**W. S. Turkington (Secretary)
'Fairyhill'
Rostrevor
Newry, Co. Down
BT34 3BB**

Plant

In typing in Plant Life by Colin Bates (QL World, March 1990), I noticed an error in line 740 where the decimal point at the beginning of the section

$x1 = x + \cos(n) * 11$

should be eliminated. Also, line 660 is superfluous, as also $0=0$ and $ot=0$ in line 780, and $le=1$ in line 1060. There variables do not appear elsewhere, and were presumably left over following editing when the program was being written.

Plant Life is a nice program and would be good as a display. Once can of course get it to repeat over and over by revising line 820 to:

820 NEXT tim: PAUSE
100:GO TO 170

or something similar.

One other point: it is often difficult to distinguish the letter l from the number 1 in some printouts. I think programmers should avoid the letter l as a variable, because of the possible confusion in reading. While the distinction may usually be made because of the logic of the code, this is not always the case and the programming error resulting from the substitution of l for 1 or vice-versa may not show an error message.

I greatly appreciate QL World and remain hopeful that some substitution for the QL will appear one day. QDOS and SuperBasic are so very much better than any rival, even six years after their inception.

**Frank Gutteridge
Corsier
Switzerland**

Fractal Report

Fractal Report is now on issue 9 and my Sorcerer has been discarded for a PC running Wordperfect 5.1, with a laser printer bought for speed and to produce a more professional result for the newsletter. Our sister magazine Longevity Report is now entirely re-typed, although Fractal Report uses camera-ready articles still, in order to reduce errors.

We have marketed three examples of fractal music, a fractal video and as a long-term project plan Fractal Report — The Video, one entry for which has already been received. Both Fractal Report and Longevity Report have exchange subscriptions with similar publications worldwide, which widens the range of all the periodicals concerned. The latter has also received an extremely positive write-up in Here's Health, as well as having acquired Simon Goodwin as a subscriber, and hopefully one day as a contributor.

Of course, in commercial terms the circulations of Fractal Report and Longevity Report are derisory, but it does show how desk top publishing can allow minority groups to form and discuss ideas, which before the technology was available, would be very isolated.

**John de Rivas
West Towan House
Porthtowan
Truro
Cornwall
TR4 8AX**

Editor's comment: It's nice to know that Simon plans to be with us all for as long as possible.

Question . . .

Editors Comment: The following letters should have been published together some time ago, but we had problems collating them. I hope they are still useful to Mr Ross and anybody in a similar situation, and thanks to Chris Wilson at Harvar PR, and Rob Pepper at Star.

I have only had my QL since February and only just bought a Star LC-10 printer. I have great problems trying to get translates to work. Can anyone help me?

**Graham Ross,
Largs,
Scotland.**

. . . Answer

With Quill, translates are provided to enable the printing of characters that have been allocated different codes on the QL and the printer. A classic example of this is the £ signs. On the Star LC-10 the £ can be obtained in two ways. You can either select the UK character set and send a code 35d to the printer or you can select the IBM character set #2 and send a code 156d to the printer. The QL uses codes 96d to represent a £. The selection of UK character set and IBM character set #2 can be made either by dip switch or using software controls. The most universal and useful way to access the pound sign is to firstly set all dip switches to ON, select IBM character set #2 via software, then send a code 156d. So to TRANSLATE the QL's £ into a Star LC-10 £ you would insert any one of the following lines into the Psion install__bas program:

1, "£,ESC,"R,ETX,"#,ESC,
"R,NULL
or
2, "£,ESC,"t,"1,156,ESC,
"t,"O

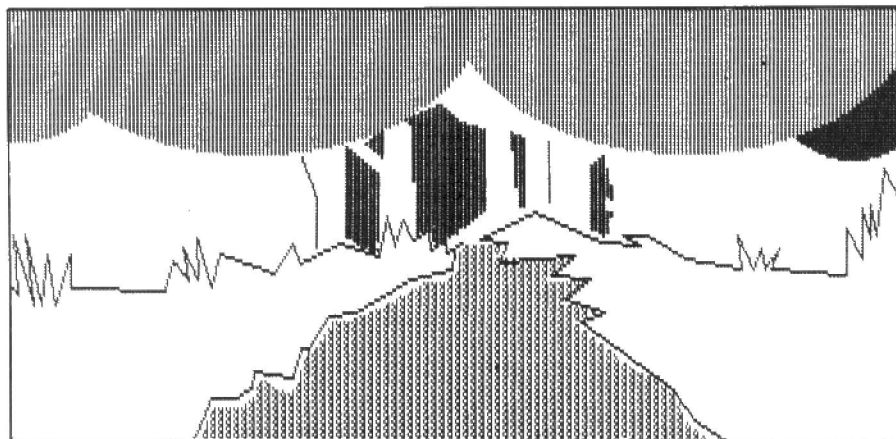
Method 1 send the codes ESC R 3 # ESC R O to the printer when a £ is found in your document and method 2 sends ESC t 1 156d ESC t O. Method 1 first selects the printer's UK character set which replaces a # with a £, prints a # then returns the printer to USA character set. Method 2 works in a similar manner.

The above system can be used to translate any of the QL's characters into any of the printer's characters.

If you do have any further questions regarding Star products, you are more than welcome to contact our Technical Support Line on 0494 471111.

**Rob Pepper,
UK Firmware Development
Engineer,
Star.**

SOFTWARE FILE



YOU STAND ON A PATH LOST IN THE DENSE WOOD OF YOUR UNCLE'S ESTATE.
YOU CAN ONLY GO FORWARD.

UNCLE LOONIE'S LEGACY

INFORMATION

PROGRAM: Uncle Loonie's Legacy
PUBLISHER: GCH Services
PRICE: £8 DISK £10 Microdrive + 10% for p&p etc. For the standard 128K QL.

This program is described as a 'puzzle adventure' and is certainly a breakaway from what one would expect from a normal adventure game.

It was written using the Adventure Creation Tool (ACT) by Digital Precision, which has helped to ensure that the program has quite a professional finish to it as with most (if not all) of CGH's commercial programs. The manual that comes with it is only four pages long and contains a background story plus the warning that "If you have bought this game thinking it to be a typical adventure, then you're in for a surprise." The manual serves as a nice little introduction to the humour used in the game but does not really prepare you for what is to come.

It seems that your long-lost uncle died leaving a large fortune hidden somewhere in his self-made ornate mausoleum. Finding the mausoleum is no great difficulty, since the game itself contains very few locations. Once there, the similarity with an adventure ends. It seems a rather pointless part of the game, actually, but I daresay that it helps to provide some confidence in the player, who is led into a false sense of security, believing it is all too easy with only five movement commands necessary to remember (left, right, forward, in and out).

However, once inside the mausoleum,

this illusion is soon shattered. There is a grand total of four locations inside the building, two of which are obviously not worth bothering with until later in the game, leaving only the entrance to the one location in which most of the game's action takes place.

Here, you are faced with a room without any exits, and a wall full of names in boxes. Entering one of these names presents you with a puzzle corresponding to that name. These puzzles range from the quite simple (naming a prime minister), to what would seem nearly impossible ("what's in a name", where all there is on screen is a boy with a speech bubble full of names), to the very intriguing (a blank screen with nothing in it).

Some of the puzzles are hidden from the player by different obstacles and nothing can be achieved unless you find some way of removing that obstacle. At first it may seem impossible to do anything about this since there are no objects lying about which may help you in your task. However, you will soon find that as you solve some of the puzzles, an object or little creature may appear which may be of assistance.

Answering each puzzle will reveal part of a picture on the main wall, and you

Rich Mellor gets his mind looked at, and discovers a Mensa-like intellect.

cannot complete the overall adventure until all the puzzles are solved.

All of the puzzles do have quite logical solutions and seem similar to Mensa-type IQ tests to some degree. They may involve adding and subtracting numbers, spotting sequences or solving anagrams. However one or two of the puzzles are a little spoilt by minor errors, which make them harder to solve.

There are good graphics to represent each of the locations in the adventure part, and the puzzles themselves are all very well laid out. The puzzles are presented as a picture on the screen which generally only has one or two words as a guide to what you are looking for. The pictures certainly add to the fun, but not all of them are helpful in solving the puzzles.

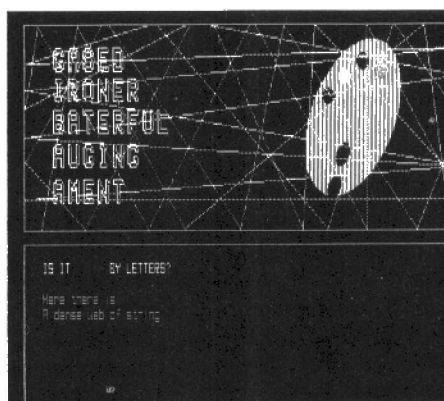
Once you manage to solve all the puzzles, you can proceed to the other two main locations, whereupon you will come face to face with a monster for whom you

should now have a name, and a computer which is waiting input. Once you have solved these two puzzles the loot is yours!

Although traditional adventurers may be put off by the lack of locations, the puzzles given are widely varied and should keep you occupied for several

days if not weeks. There are some other intriguing little features in the game, such as the ability to say 'monster' which makes the monster appear wherever you are. You could shoot him if you wish, but unfortunately this has no bearing on the final outcome of the game — still, I guess it provides a means of releasing your frustration when you cannot come up with a solution to one of the puzzles.

Overall the game feels like aerobics for the brain and even if it does lack lamps, magic rings and goblins, you feel a whole lot better for having figured out the answers. The presentation of the puzzles should appeal to most traditional adventures. It is perhaps one of the few games for the QL which could prove educational and fun for the whole family.



QL SCENE

PDQL ROW CONTINUES

The dispute between Miracle Systems and PDQL, reported in last month's *QL World*, remains unresolved as we write.

Stuart Honeywell of Miracle systems confirmed that, so far as Miracle Systems was concerned, PDQL were 'still in default' in mid-June, a month after our previous report. "Our solicitor is preparing to serve a statutory demand on John Silk now," he said.

EEC Ltd did not in the event join with Miracle in the action. "I haven't even seen my solicitor yet. We were going to go in with another supplier, but they went ahead. They had much more money involved. I suppose they wanted to get in first, which is understandable" said EEC's Bill Richardson, confirming that Miracle Systems

was the supplier in question. "I understand PDQL paid up," he added. Stuart Honeyball however stressed that his own action was still in progress and that he was not the source of Bill Richardson's information.

Honeyball added that Bill Richardson appeared to be 'dithering' about arrangements to join their action, so they went ahead without him.

Richardson is "Still owed money and I will continue to talk to him (John Silk) about it." Will Richardson be taking up a court case on his own behalf? "It's a matter of time and money. The county court is an utter bind and there are probably more useful things I could spend my time on, in court terms. I may do" he said.

Tony Tebby quits UK

Peter Rowell

A barbecue was arranged by the Quanta Mid Anglia Subgroup to thank Tony Tebby for his support to QL owners in the past and to wish him and his family every success in their future life in France. About 30 people attended on the evening of a beautiful summer day, Saturday 16th June.

Sid Humphreys, the Quanta Chairman, presented Tony Tebby with a crystal wine carafe and four matching glasses. He thanked Mr Tebby, on behalf of Quanta, for the valuable support and software which has been available to QL owners in general, and Quanta members in particular, for the past six years. He wished him a happy future in his new home, hoped he would remember his

many friends in Quanta and that we would see him at workshops, both here and abroad. Mr Humphreys then presented Mrs Tebby with a bouquet.

In reply, Tony said it was unusual for a dealer to receive a gift from his customers — it was usually the other way round. He appreciated the gesture and as he was going to live in a wine district in France it would bring to mind his many Quanta friends as he enjoyed the local vintage. He intends to visit Quanta Workshops and Shows whenever possible.

Tebby software will be obtainable from Care Electronics of Watford, who have handled many of his products in the past.

Athene moves on Turboquill+

Following recent reports that some users had had difficulty in obtaining delivery or other help with *Turboquill* + from the publishers, Athene Computers, *QL World* learns that TK Computerware are now the official distributors for the program, with sole rights to duplicate and sell it.

Proprietor and programmer Julian Dyer admitted that the company, which has diversified widely into other software, had had problems with hardware breakdowns which had prevented them from duplicating programs to meet orders in recent months.

"We were getting no orders for *Turboquill* and we thought people had lost interest, so we stopped maintaining our QLs", said Dyer. "But then we heard

that somebody else was selling it; we are not sure who was involved at the moment."

A revival of interest in the updated version, *Turboquill* +, was generated by a favourable review in the March 1990 issue of *QL World*, which gave Birmingham firm PDQL as the supplier. The program had appeared in both TK Computerware and PDQL's advertising for some months prior to this, although PDQL ceased to advertise it in the following issue.

Julian Dyer had not been following events in the larger QL community, and confessed to being surprised when a colleague showed him the review "quite recently". "We have had no orders from PDQL for *Turboquill*. He bought five or

ten from us in person at the last Microfair we attended, which I believe was last summer," he said. *QL World*, indeed, has received among others complaints about non-supply of the program by PDQL, from readers ordering the program from March 1990 onwards.

Julian Dyer added that he would like any Athene customers who had had problems in getting the upgrades to contact him direct, but that TK Computerware would now be the sole authorised distributor. "It is difficult to tell if there are pirate copies," he said. "Because the program is not copy-protected, but originals sold up to now should have a characteristic blue sleeve, printed by us."

Existing Athene customers

should contact them on 0705 511439.

Quanta

The June 1990 issue of the Quanta magazine has articles on dedicated function keys for hard-wired keyboards, the Amiga QDOS emulator, embedded codes for *Quill* and the FX-80 printer, controlling share investments with *Abacus*, plus group news and shorter contributions from members. We note that *Text87* is being used to produce the newsletter, with the help of a printer driver supplied by Fred Toussi of Software87.

T



TROUBLE

A

P

R

O

B

L

Bryan Davies switches from games to spelling, and looks at buyers' problems.

There are many facets to the overall QL Scene, and the average user may be quite unaware of most of them. One reason for this is that some of the people heavily involved with the QL make relatively little effort to publicise their activities. Apart from (maybe) sending occasional news items in to the editor they appear to feel that there is nothing else they should, or could, do to "spread the word". This seems a bit shortsighted, especially in view of the inevitable shrinkage of the market as the years go by. What I am suggesting is that programmers, hardware developers, writers of newsletters etc, consider sending details and samples of their work to *each* of us who write regularly on QL matters. We can't (shouldn't) make comment on things we don't know about; it can be unwise to simply quote someone else's comments about a product.

An upgraded version of the spelling-checker *Spellbound* is scheduled for release fairly soon. Note has been taken of various criticisms directed at the (very successful) original version, and changes have been made to deal with several major points. Files will be checkable retrospectively, without the need to use *FileBound* as well. The checking speed when used in this mode will be much greater than was achieved with *FileBound*. Loading of the dictionary file will be faster. If the dictionary in memory is written out — eg to save it part-way through a session — it will still be maintained in memory for further use, avoiding the previous problem where all one achieved by writing it out was fragmentation of the freed-up memory so that it was not usable by most other programs, without being able to do any more checking.

The switching on and off of the spell-check function will be more intelligently controlled than previously, which should come as a welcome relief to those like me who found the restrictions placed upon keyboard actions impossible to bear. While thinking of Sector Software's offering, some users may not be aware of the

significant improvements that *Files 2* has over the original version. One irritating feature which has been modified is the numbering of files, which kept changing as you performed actions that changed the overall number of files on a medium. You now know that, if you delete file 1, another file has not been numbered 1 in its place. A weakness compared to Ice was the lack of a View function, to allow you to inspect the contents of a file, but one exists in *Files 2*. Files can be selected by typing-in the first few characters of the file name; the more characters you type, the more accurate the selection. Typing <LE> might choose LEB, LETS_BAS, LETTER_DOC, LETTER1_DOC etc. For users who are not convinced they need the full set of function provided by *Taskmaster*, *Files 2* can be run on its own.

Games players might have seen adverts for a newsletter called *QL Adventurers' Forum* (QLAF) and have wondered whether or not to risk sending off money for it. Until recently, I knew nothing about the newsletter, but a copy of Issue 9 has been loaned to me and I have had the chance to see what it's like. Basically, it is a collection of comments and reviews of games programs — it is a very specialist journal — but it also contains some general QL news. It is generally well written and laid out, and the printing is decent. With 36 A4 sides, it is a similar physical size to *QL World*. There are a few adverts. To a large extent, it is dependent upon reader input, being a discussion forum for games players. The newsletter has been renamed *QL Leisure Review*, to reflect the fact that the contents are no longer solely concerned with adventure games, but cover the full games spectrum. As you can buy single issues (I understand Issue 3 of the Review is out), it should be worth trying (see INFORMATION below).

It is surprising how few people call or write to point out errors that creep into articles. There has been a fair amount of internal discussion on the subject recently, and we trust there will be fewer readers wandering around with bemused expressions in future. One boob that has been pointed out was the result of (hopefully) temporary brain-fade on my part; an *Eye-Q* screen save is normally the full 32 KB, as with the SuperBasic SBYTES command, and it is only the compressed screen (typically 12 KB) which cannot be reloaded with LBYTES. The compressed screen needs the *Eye-Q* SB extension ALOAD to put it back on the screen. *Professional Publisher* loads 32 KB pictures, including uncompressed *Eye-Q* screens, with the "SBYTES

screen" option; compressed *Eye-Q* files are loaded with the "Eye-Q screen" option.

READERS LETTERS

No answer has been received from PDQL concerning complaints from D. S. Barnes, J. Wass, J. Hay, A. Munden, B. Harrison and D. Armstrong, but A. Munden has written to say he has been trying for four months to get a copy of *Spell-Bound* from PDQL and, despite repeatedly 'phoning and writing to them, he has not received it; the cheque for it was cashed promptly. A similar story comes from H. Graham, who has been waiting "105 days" (as of May 16th) for a disk drive and interface ordered from PDQL. Again, the cheque was apparently cashed promptly.

Norman Larkin reports receiving his QL and keyboard back from Schön, both apparently working. D. S. Barnes has now received printer ribbons ordered from TK Computerware; a second lot were despatched Recorded Delivery.

Overseas buyers obviously have a bigger problem when something goes wrong with an order, and two letters received in recent months are typical of the type we get at fairly regular intervals. A word of caution, though; it is very easy to assume the fault lies totally with the supplier ("them" rather than "us"), but I have found that further probing can show that the buyer brought the problem on him/herself. That is not to suggest that this is the case here, however. James Pollitt wrote from New Zealand to praise Miracle Systems for their "excellent products and Airmail service to match". He went on to give a "black mark" to Sector Software for taking about five months to supply a copy of *KeyDefine* and a test lead, and not responding to his enquiry about adding memory to a Thor I. He also said a letter asking what was happening to his order was not answered. Apparently the goods had been sent, nominally by airmail but with insufficient stamps to pay for that, so they had gone by sea (12-14 weeks). Regarding the Thor enquiry, I'd be surprised if any supplier could deal with it now. That's not the end. The next black mark was for Schön PCP, who apparently banked an £80 cheque for a KBL casing and flexicable in October 1989 but the goods had not been received by Pollitt as of 3rd March 1990. J. Roy Goodall wrote from Belize (Central America) to say he had never got what he subscribed to SUB for; only two issues of the "monthly magazine" reached him. He gets no output to his printer from a copy of

SHOOTER

E M S O L V E D

TechniQL, bought from T K Computerware, and asked them for assistance, finally sending the program back (apparently for TK to set the printer-driver up for him). The matter had been going on for "over one year", as of March 1990.

Next in line for a rocket were Miracle Systems; he asked them (and enclosed a cheque) to supply a new Serial-Parallel printer interface, and quote for repairing another one, in December 1989. Two chase-up letters were sent, but no reply received as of late March this year. Finally, he berates Transform for sending him an "Organiser-to-Archive 1" disc, and also for not supplying a "six datapak case to replace the one sent by mistake and returned to you."

As indicated above, if one knows the whole story in such cases, the impression gained may be different from the obvious one, but there is the constant thread of unanswered correspondence. My own experience fully supports this. Suppliers hate answering letters. They regard the job as of relatively little importance in the

business of selling things. In this, they are only behaving as most people do in daily life, surely? Do you reply to letters promptly? Indeed, do you write letters at all, or do you only correspond by telephone? Personally, I don't much like the 'phone, and can sympathise with suppliers who are not going to incur hefty 'phone bills dealing with all enquiries that way, but — to run a good business — you have to respond to buyers one way or the other. Making comments like this in the past has brought me outraged calls from some suppliers, so please note that I have (almost) always found both TK and Digital Precision to be very prompt in replying to my enquiries! With most suppliers, reponse has varied considerably.

Rog Cox points out that it is possible to avoid using all of Quill's 10 translate entries for individual printer functions (such switching-on condensed printing) by using one keyboard character to pass the ESC code to the printer, and following this character by the others needed at particular points in a document. His

translate for this reads <128,27>, 128 being the decimal code for a (the German umlaut "a") and 27 being the code for the ESC which prefaces most code sequences used in Epsoms-compatible printers. That is, at the point in a document where he wishes to send a signal to the printer, he types the a character — which sends ESC to the printer — and follows it by what ever the printer expects to see after ESC to produce the function required. For example, ESC"E" typically sets Emphasised printing on. I leave it to you to think out just what you type into the document. This method is not recommended for users with no technical knowledge!

INFORMATION

QL Leisure Review, £1.50 plus 10% post and packing per copy, or £6 inclusive for ten issues:

C.G.H. Services
Cwm Gwen Hall
Pencader
Dyfed SA23 9HA.

QL SUPERTOOLKIT II by Tony Tebby

Over 118 Commands:— Full Screen Editor, Key Define Print Using, Last Line Recall, Altkey, Job Control, File Handling, Default Directories, Extended Network.

16k Eprom Cartridge Version @ £ 24.15d
Configurable Version on Microdrive @ £ 24.15d

MIRACLE SYSTEMS PRODUCTS

QL Trump Card 768k (Toolkit II etc) @ £251.85b
OK Disc Interface (inc Toolkit II) @ £ 99.82b
QL Centronics Printer Interface @ £ 28.75d
QL Expanderam 512k ThruCard @ £ 99.99e

QL HARDWARE

Single 3.5" Disc Drive & (Own PSU) @ £103.50a
Dual 3.5" Disc Drive & (Own PSU) @ £188.60a
3.5" DS/DD Discs — 10 off @ £ 9.20c
Q POWER REG. The only real solution to your QL overheating (switched mode power supply run cold) @ £ 24.15c
QL Keyboard Membrane @ £ 11.50d
QEP III Advanced Eprom Programmer @ £121.90d
Care Eprom Cartridges each @ £ 5.75c
ULA Chip ZX8301 @ £ 15.64c

PRINTERS

STAR LC10 Mono @ £179.40a
STAR LC10 Colour @ £224.25a
CITIZEN SWIFT 24-pin Colour @ £368.00a

SOFTWARE 87 (State MDV or Disc)

TEXT 87 V.3.00 @ £ 60.00d
FOUNDED 89 @ £ 15.00c
FOUNTEXT 88 @ £ 25.00c
TEXT 87/FOUNDED 89/FOUNTEXT 88 @ £ 94.99b
2488 PRINTER DRIVER @ £ 15.00c
Upgrade to Text 87 V.3.00. Return old copy together with @ £25.00d

MONITORS (Price including lead)

Philips BM7522 Amber Hi-Res @ £ 97.75a
Philips CM8833 Colour Med-Res @ £253.00a
Philips AV7300 TV/tuner for above @ £ 69.00b
Philips BM7502 Green Hi-Res £97.75a

HOW TO ORDER:

ALL PRICES INCLUDE VAT

By Post. Enclose your Cheque/PO made payable to CARE Electronics.
Or use ACCESS/VISA. Allow 7 days for delivery

TONY TEBBY SOFTWARE (QJUMP)

QFLP (Micro/P disc interface upgrade) @ £ 14.95d
QMON II @ £ 23.00d
QPTR Pointer Interface m/drive @ £ 36.80d
QPTR Pointer Interface + 3.5" disc @ £ 29.90d
QTPY Type/Spell Checker @ £ 29.90d

ZITASOFT SOFTWARE by Steve Jones

LOCKSMITH copies M/DRIVE — M/DRIVE @ £ 14.95c
4MATTER + LOCKSMITH copies M/DRIVE — DISC @ £ 23.00c
The above programs are not for use in the UK.
SHRIVEL memory shrink prog user definable ie 128k or 192k or 256k etc @ £ 13.80c
TOOLCHEST utilities to allow the creation of customised mdv doctor prog £ 14.95c

SIDEWINDER — High resolution printer driver prints full screens or parts of screens from postage stamp size to large banners. Prints sideways, invert, scale, mirror, text inversion....
@ £ 19.95c

SIDEWINDER PLUS — (for expanded QL's) includes all the features of above, PLUS multiple label printing, desktop publishing files and printer driver for 24" pin plus LC10 and 3x 80 colour printers. (Please state 3.5" disc or M/D)....
@ £ 23.00c
Upgrade to Sidewinder Plus @ £ 8.05c

HEAT TRANSFER RIBBONS & PENS

Print your own T-shirt Design. Colour or black and white. Please phone for further details.
Star LC10 NX1000, Rainbow £ 17.25c
Star LC10 NX1000, Black £ 11.50c
Okidata ML80/Epson FX/MX80/Epson LX80 Black £ 11.50c
Citizen 120D, Black £ 11.50c
Epson FX100, Black £ 11.50c
Jumbo 5 Colour Pen Set £ 17.25c

READYMADE LEADS

RGB QL to Phono @ £ 5.75c
RGB 8-6 pin DIN @ £ 7.13c
RGB 8-7 pin DIN (Hitachi) @ £ 7.13c
RGB 8-7 pin DIN (Ferguson) @ £ 7.13c
RGB 8 pin to SCART (Euro) @ £ 11.50c
6-way PCC 25-way 'D' (Printer Ser 1) @ £ 9.89c

QPAC I Desktop accessories. Calendar, alarm, calculator, typewriter, digital clock, Sysmon @ £ 21.85d
QPAC II Main menu windows adjust automatically to size. Files, directory, view, print, delete, backup, jobs, pick, Rjob, sort, channels, things, exec, wake, buttons, Hotkey, Hotjobs. Fully multitasking, allows many programs to run at once
Requires min of 256k expanded memory @ £ 49.91b
Upgrade QRAM to QPAC II return master @ £ 29.90b

CARE ELECTRONICS
OPEN
9am-5pm Mon-Thu
9am-4pm Fri

800 ST ALBANS ROAD,
GARSTON, WATFORD,
HERTS, WD2 6NL.
Tel: 0923-672102
Fax: 0923-662304

Please add carriage
a=£11.50 b=3.45
c=£1.38 d=£2.30

What is *Fleet Tactical Command*?

Very briefly, *Fleet Tactical Command* (FTC) is a 3-D, real time, Naval wartime scenario simulator, set in a 1,000×1,000 mile plot, that can be played over a network or run on a single machine for practice. It will primarily be of interest to an adult market.

Each player has 16 independent Warship and Support ships. Within the confines of the plot are jetties, tide flows, tide changes, depths, shoals and so on. The aim of the game is to enforce an exclusion zone in the area known as 'Medusa South West' by seeking out and destroying the enemy.

FTC has in my own opinion two major advantages over almost any other computer game: first, two players on different machines can pit their wits against each other in a real time, fully inter-reactive scenario. Secondly, FTC encompasses most aspects of Naval strategy and warship operations from navigation through to gunnery warfare and even damage control routines. This puts FTC in a class above almost all other Naval orientated simulators in that as far as we are aware, all other simulators, both professional and domestic, tend to relate to single, very specific operations such as gunnery, warfare, manoeuvring, etc.

The program will run on any version of the QL with or without expansion memory and does not require any toolkits to be fitted, even when using the QL-QL network.

Although developed on the QL, the program, over a period of time, will become available on other popular computers (as I write, re-coding for DOS-based machines is currently underway). The aim is to allow a combination of two different computers connected via serial or modem links to run the program. Martin Florichs of Olympic, West Germany, recently tried to run FTC on the Atari ST with a QL Emulator, and reports, I quote "there seem to be no problems".

FTC, you may not be surprised to hear, is derived from the children's game of Battleships. Back in Christmas 1986 we programmed a very simple version of the game to work over the QL-QL network. The kids loved it! So enthused were we by the success of this that we

FTC The Story

Authors Robin Barker and Andrew Hopkins tell the real-life yarn of Fleet Tactical Command

commenced programming a more sophisticated 2D version with shaped blobs moving across the display.

Development problems then began to abound.

For programs that require constant data transfer a major problem with the much maligned but in fact very useful QL network exists: the QL network driver is designed to send data in 256-byte blocks. In practice, this means that if you wish to send less than 256 bytes of data across the network there is no legitimate way of sending it without closing the output channel.

The Qdos FS.FLUSH Trap (which is called by TK2's FLUSH command) does not work on the QL-QL network driver (NETO__ Channels). Continual opening and closing of channels takes time and to compound the problem it is not long before the System Variable channel count reaches its maximum value, at which point it is no longer possible to perform channel open/close operations.

Eventually we found that the trick is to fool the network driver into thinking that it has received its full quota of data by poking the driver's offset NET__RPNT. Anybody trying this will have to take into account that TK2's network driver is substantially different to the Qdos Version.

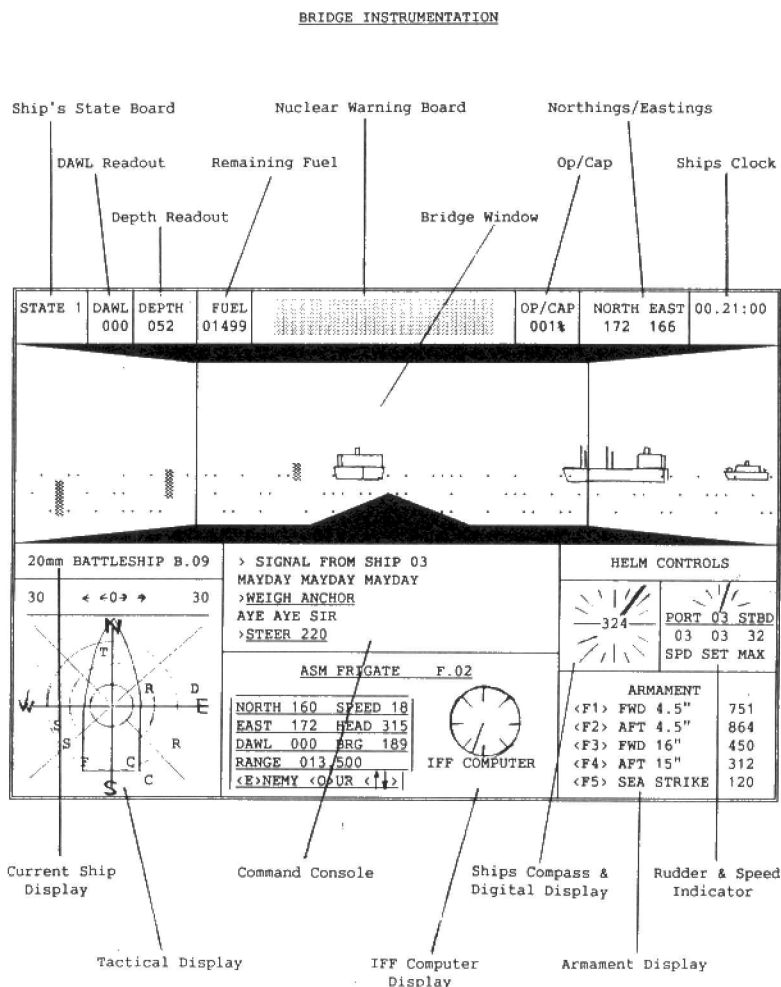
The 2D Version continued to grow in complexity. Along came DP's *Turbo* compiler. This appeared to be the solution to what was fast becoming a huge, chronically slow program and so it was, until we decided to draw the ships in 3-D images.

In an effort to speed up the program we heeded advice given in DP's *Turbo* documentation. Integer variable loops and pre-defined floating point tables were used wherever possible. However, with every possible optimisation performed, even when compiled, the 3-D version of the program was too slow to be interesting, with a code size too large to be practical.

Due to a combination of programming problems and authors' commitments, FTC development was shelved in 1987.

We eventually took another look at FTC in 1988 and made the decision to completely re-write in pure machine code, a massive undertaking. Code execution speed was all-important. As a matter of interest when networking, the game now runs at some 33-38 times faster than 'normal real time'.

Initially the bridge window, tactical and all analogue displays were drawn directly to the screen using Qdos LINE and BLOCK commands. It was necessary using this system to store all drawing plot variables for previously drawn ships to allow them to be 'blanked' out without totally destroying the display. Unfortunately the Qdos LINE command is inaccurate. As a consequence debris used to remain on the display after blanking routines.



The BLOCK command also has problems. Try this 'BLOCK 514.514.0.0.7'. Given that the maximum window parameters are 512x256 you would expect an 'out of range' error to be generated. Not so! It would appear that the routine MODS its supplied width and depth parameters by 512. This particular bug proved less than helpful.

These problems coupled with slow Qdos code execution speed for the BLOCK and especially LINE commands eventually forced us into writing the program's own BLOCK, LINE, and as an extension PRINTING routines. The new routines use exactly the same calling parameters as the original BLOCK, PRINT and CURSOR calls. The LINE command now uses pixel co-ordinates.

The main advantages of the new code are that the routines do not have to make the same laborious parameter checks as do the Qdos versions. The routines are optimised for maximum execution speed and finally have the very major advantage of working not only on the screen display but anywhere in memory.

Being able to draw the Bridge Window display in memory meant that we could dispense with 'blank-out' variable storage. A CLS is performed every program loop to the memory version of the Bridge Window display which is then completely redrawn. When complete, the memory version of the display is overlaid 'quicker than the eye can see' to the screen display.

Interestingly the Tactical Display background is copied from the screen into memory when the program is first loaded. This is then copied to the display build-up area instead of performing a CLS thus saving the program the effort of having to re-draw the background for every Tactical Display update.

With the exception of the Command Input Consul which is a low priority task, the display is now completely built up by the program's own routines.

Drawing the 'moving sea' effect proved to be a major problem. SV.RAND does not update quickly enough to be able to use its data directly.

Solution

Eventually a solution was found that involved using SV.RAND to point to a location in the first 255 bytes of memory. By some tricky ANDing and ORing of consecutive data lifted from these locations a very passable moving sea effect was created, as users of FTC will probably testify.

Development continued apace. Myself and Andrew (FTC's co-author and mathematician) spent many hours playing the game on debugging exercises. (The last game before the final release lasted for some 11 hours). Debugging FTC quite frankly is a nightmare. The assembler listings for the program are now in excess of 500K.

Very close to the release date and inconveniently after advertisements had been lodged with this magazine, it was found that even after the most careful calculations the program would not run on an unexpanded QL.

Due to the complexity and the interactive nature of the program this took weeks to solve. The end result is that the program 'just' squeezes into a 128K machine.

To enable File saving operations to take place it is necessary to release the portion of memory holding the Tactical Display background copy. This is achieved by 'saving' the picture on the screen display during file operations. Once file operations are complete, memory space is re-allocated and the tactical display copied back from the screen.

Unfortunately due to lack of space and for technical reasons a modem option on V1.01 could not be implemented. It is envisaged however that V1.02 (free to registered users of V1.01) will contain the option.

The program uses the QL's multitasking capabilities and runs as two distinct tasks: the first task, which has a very low priority, handles all keyboard input and displays signals, etc. The second, high priority task has a lot to contend with. To describe just a few functions for every main loop:

Network input/output control and data distribution.

16 ships position updating according to speed, course, tide flows, etc.

Up to 496 possible collision checks (if networking the current ship has to check its distance from up to 31 other ships).

Ships' relative bearings against each other, required for Drawing the ships and IFF computer information.

If within the 25 mile 180 degree display range, if daylight or near a jetty, drawing the ships.

Updating IFF, computer and tactical displays, fuel consumption, ships' stores consumption, missile and torpedo tracking, damage control routines, etc.

Check for minefields, groundings.

Plot jetties, mines, buoys so on and so forth. . .

For System Variable and Screen display compatibility the program re-defines its own parameters upon installation, using information acquired from MT.INF and SD.SCRB respectively (system variable, and screen display addresses), compatible with Minerva roms. Initial parameter redefining speeds program execution. For the uninitiated all this means is that if you have a section of program code that requires for instance the value of SV.RAND, instead of saying something like `SYSTEM__VARIABLES=MT.INF` (the trap to return the System Variable address), use `IF (SYSTEM__VARIABLES+SV.RAND)>20 THEN JUMP TO XXX`. That particular piece of code is POKEd with the direct address of `SYSTEM VARIABLES + SV.RAND`

found during installation procedures, saving time when the program is running.

What is not likely to be common, knowledge is that the system variable SV.RAND is NOT a random number generator but is in fact a sequential frame count (50 frames per second in UK).

Within certain practicalities every effort has been taken to keep the program as realistic as possible. Fairly close to realism is the way the ships manoeuvre, the ship replenishment at sea routines, gunnery ranges (with a degree of licence) and damage control routines.

Navigation

Navigation is the major exception. It was decided that it would be very impractical for the laymen to navigate using Latitudes and Longitudes. Therefore the much simpler grid system is used, ie the plot starts at 0/0 on the bottom left hand corner incrementing horizontally and vertically in sea miles. (Navigational aids and charts are supplied with the package.)

FTC is such that it would not be difficult to convert it to a fully comprehensive professional package. We have had some interest from professional users.

Four QLs of varying degrees of reliability have been used in the development of this program. Contrary to what appears to be popular belief, the QL-QL network works well on most machines. If yours does not then have it seen to!

The program was developed using this network, SERIAL link communication options only being added later. The game can last 12 hours, during which time we have found the network performs faultlessly.

It is our experience that on some machines the QL-QL network using TK2's file servers tends to seize up if not used fairly soon after starting the machines. However, if you open NETO__ and NETI__Channels no problem appears to exist. Perhaps somebody can explain.

When playing QL-QL we recommend that the QL network facility is used. When running it is perfectly reliable and SOUND is only available if using this option. Unfortunately the QL cannot generate sound and use SERIAL links at the same time. Talking of sound, the NUCLEAR ATTACK IMMINENT alarm sound is in fact derived from the film "ALIENS" — This time it's war !!

From our point of view QLs are ideal development tools. Not only are they cheap, they are more versatile, even now, than many other far more expensive machines. (Microsoft windows? who needs them when you have the QL!)

It has taken 2 to 3 years to develop FTC and has been a very expensive, mind-crunching exercise, one that is not likely to be repeated. Development costs of high quality games considerably outstrip utilities that cost twice as much.

So why did we do it? It's a Passion Fulfilled.

Bryan Davies launches into a serious new game.

(081)-291 3751

behaviour was traced to the presence of an Ice rom in the test system, interacting with the copy—protection of FTC. It did not affect use of the program, and the suppliers are now dealing with the matter: it is by no means unknown for Ice to cause hiccups in programs, as it seems to believe it has first priority over everything.

The program runs on a basic QL, which is quite an achievement. Roughly 84 KB is used, and the disks/microdrives are not accessed for other than loading and saving purposes once the program has started. You, and your enemy, have 16 ships each, all of which can be in quite advanced stages of disintegration, and that means a lot of parameters, as constant checks have to be made on such matters as ships' locations and proximity to each other and natural hazards.

There is so much detail to the program that it is not possible to do more than skim the surface here. Odd notes in the instructions give clear pointers to the depth of thought which has gone into the preparation. For example, firing a torpedo from a submarine when proceeding at full speed can result in the submarine overtaking the torpedo! Loss of controls in an emergency may be dealt with in some cases by giving the order to connect-up reserve cables and switch on an auxiliary power supply generator. You need to remember a special code to be permitted to use nuclear armament. Keeping the crew at Action Stations for lengthy periods (or running out of food) is likely to result in mutiny and sabotage activity, so the men need to be stood-down at intervals.

Spaté

Programs do not always come singly — you may get a sudden spaté of messages, advising that several ships are suffering from malfunctions, fires etc. When steering, you have to bear in mind any local current flowing, or you may never succeed in berthing at a jetty, coming alongside a tanker for refuelling, or lining-up to fire at a target. The bulk and weight of ships is taken into account in the way they turn and get up speed.

Rather than giving a blow-by-blow description of the operation of the program, and then not recommending buying it, let's be clear first that I think the program is good, and well worth consideration even by non-games players. It is well thought-out, and solidly executed. Being totally machine-coded, it does not suffer the delays exhibited by some programs not produced by mainstream software suppliers. The graphics are good and the display changes are snappy. That's not to say that the ships and waves you see are like photographs — they are made of lines and dots as usual, but presented imaginatively. Above all, the program is interesting to work with. Something is always happening on-screen, even if you are merely drinking yet another coffee. It is not essential, but

preferably, to use two QLs, and to have two players.

You are not playing against the computer, but against a person. This does not mean the program merely takes account of your orders. It throws in a few diversions to keep you on the ball. You need to react to everything; this is one of the prime attractions of the program, raising it well above the "dumb" level. Each ship can be controlled directly (you are "on" it) or by means of signals. Commands can be entered as text, or via cursor or CTRL key combinations. Real-life features are included, such as varying water depth, armament, damage-control, steering and speed control.

To avoid undue complication for the user unfamiliar with navigational terminology, latitude and longitude are replaced by a simple grid divided into distance in miles, with the lower left corner being the 0,0 point and the top left 1000,1000. A displayed ship location of 458 East, 566 North means the ship is roughly in the centre of the main chart.

Five area charts are provided (four copies of each) with white, light blue and medium blue colouration being used, along with contour lines, to indicate water depths. The charts have mile and kilometre scales, a compass "rose", and water current markings (eg a Gulf Stream). There are natural features (lagoon, shoal, sound, reef, etc.) and harbours, with four jetties at which ships can be berthed. The "high-tech" side has an echo sounder, a plan display or the area surrounding the ship you are on, range/speed/bearing indicators for targets and the relevant speed and steering parameters for the commanded ship.

A separate panel shows the armaments situation. The armaments include the obvious big guns, depth charges and torpedoes, and also nuclear missiles and "chaff", the metallic debris thrown up to confuse missile navigation systems. The missiles are of the automatic-homing type, relieving the player of the need for precise aiming. Once two ships carrying missiles get into range of each other there is nothing to stop each blasting the other to oblivion.

Some emphasis needs to be laid on the interconnection between the two computers; success with the program depends upon this link working properly. Before buying the program, check that your QLs

will communicate with each other, over a period of several hours. The machines can be linked via the Serial ports, although most users are unlikely to have a suitable cable available; note that the connectors needed are not the same as those used on telephones, although you can use that type after a bit of careful carving to make the keying match the QL ports. Bear in mind that the SER1 and SER2 connectors have to be wired differently.

The Network lead supplied with the QL

GENERAL

The commands to operate the programme are described in the DETAILED COMMANDS section of this manual.

Although they might appear slightly arcane at first, especially to those with no "sea going" experience, you will soon become familiar with them.

When "time" is referred to, unless specifically referred to as "REAL TIME", means GAME TIME.

PORT & STARBOARD

Looking towards the front of the ship PORT is left and STARBOARD is right.

BRIDGE WINDOW

Through the bridge window you will be able to see ships that are within 25 miles of your current position. The further away, the smaller they appear.

It is essential to grasp the fact that the bridge window displays some 180 degrees of vision; i.e. anything that is displayed to the left of the window is in fact on your left hand side.

In the bottom centre of the window is a triangular shape that represents the bow.

Should the anchor be down, a white line will be seen crossing the bow. This is the anchor chain.

Viewing angles:

It is possible to obtain views from the FRONT, LEFT, RIGHT or REAR of the bridge.

For example; by entering "REAR" a prompt displaying "REAR VIEW" will appear above the window display. On all ships except tankers you will see the stern and possibly a flagpole.

Entering "FRONT" will restore forward vision.

CHANGING SHIPS

To change from ship to ship either enter CHANGE TO SHIP followed by the ship number required, or by pressing the "ESC" key the prompt "CHANGE TO SHIP" will be written out.

Pressing the "ESC" key will restore forward vision.

NAVIGATION PRINCIPLES

Some degree of navigational ability is beneficial when plotting courses to steer. For those that have no navigational or geometrical experience, here is a layman's guide to navigation. This guide does not seek to explain the intricacies of the subject, but demonstrates a reasonably accurate system to decide the course to steer in order to arrive at your intended destination.

For this exercise assume that your ship is at 925E 550N, and you wish to "cross" to a point 550E 590N. (Eastings are always quoted before Northings.)

1. First plot a point "A" at your present position on the large scale chart. This position is shown as Eastings and Northings at the top of the chart. This position is shown as Eastings and Northings at the top of the chart. This position is shown as Eastings and Northings at the top of the chart.
2. Plot your intended destination "B" on the chart. Join these points "A" and "B" using your ruler. Measure this line AB with your dividers. With the divider's read off the distance to travel on the scale on the right hand side of the chart. In this instance this should be approximately 450 miles.
3. From this it can be seen that if you steer a course of 215 degrees, as may be expected from a direct reading using the compass, the current flow of "track" will push you further south and east than you intend.
4. Decide the speed you wish to set, with due regard for the economical cruising speed of the ship concerned and the urgency of the situation. In this instance we will set a speed of 15 knots.
5. Divide the distance to be travelled (450 miles) by the set speed (15 knots) and the result will be the total estimated voyage time in knots.
6. Multiply this 4th voyage time by the speed of the prevailing current, as indicated on the chart (in this instance 2 knots).

The result will be the "track", i.e. the total offset caused by the current on your voyage.

Time x Current = track
450 x 2 = 90 miles

Draw a line from point "B" in the OPPOSITE direction to the current flow, parallel to the current arrow on the chart. Set your dividers to the "track" distance calculated above, using the scale, and mark a point "C" UPSTREAM from "B". "C" is the course deviation.

is the obvious one to use, however. My experience using the QL network functions (usually in conjunction with *Toolkit II*) has not been wholly satisfactory, and it was no surprise to find the hours ticking away to no avail when an attempt was made to load the program to both computers from one disk drive. The instructions were not clear on the use of TK11, and the reason for the problem was that the Server (called by the <FSERVE> command) running on the QL to which the disk

drive was connected had to be switched off (by <RJOB SERVER>) to enable the program to reserve memory space. (When using a command of the form <NFS USE mdv, n1 flp>, bearing in mind that it was to be preceded by <flp USE xxx> if the QL it is typed-in on has a disk interface.)

To save further waste of time, separate program copies were used on the two machines, and there was then no problem getting going. **TK11 is not needed when the program is run this way.** It is not necessary to enter network station numbers prior to running FTC, as you are asked to enter them during program initialisation. If network commands work properly initially, but the behaviour becomes untrustworthy later in the same session, it is advisable to consider either trying other QLs, or modifying the network driver circuits (see "Trouble Shooter" in the January 1990 issue). It is said that early QLs (pre-JM) didn't have a properly-working network, but my machines are JM and JS, so I haven't checked this.

The note in the user instructions about running FTC strictly on its own is serious; even trying to run *Lightning* to see if it speeded-up the graphics resulted in reset as soon as loading was attempted (the program rewrite screen drawing routines for faster operation, anyway). For once, I did not attempt to run my normal program-switching system!

The opening screen is well-drawn, as are all subsequent ones. Attention has obviously been paid to getting everything in the right place, symmetrical, and in colours that enhance the visual impression. The operational display is a view from (and of) the bridge of a ship. The upper half is the sea, and what is on it: the lower half is the bridge instrumentation. Across the top of the screen is a narrow strip with more instrumentation.

The main panel is split vertically into three sections, with the left containing the plan display of the area surrounding the ship and giving the identity of the ship (with 16 ships each, you need a reminder of which you are on, at regular intervals). The right is split horizontally, the upper half having the steering and speed control settings and indicators, and the lower one giving the armaments situation. The centre panel is also split horizontally, the lower half having the indicators relevant to chasing enemy ships (or locating your own), and the upper half being the input area for typed commands.

Perhaps the most difficult task for the QL games programmer is producing an acceptable degree of realism and action on the screen without making the program too large to run on an unexpanded machine, and slowing user activity down to an unacceptable level. The QL screen display is not up to the standard of some more-recent computers, no matter what the programmer does. One problem in a naval simulation is giving the effect of sea motion, and this has been handled quite well here. The viewed ships themselves are simple line drawings, and they move sensibly in response to steering and speed inputs to the commanded ship. As with the simple "Battleships" games mentioned earlier, a feature of FTC is the realistic screen-wide flashes which occur when hits are made on nearby ships. An irritant, which raises the reality level, is fairly sudden blacking-out of the screen, corresponding to day turning into night, as indicated by a clock at upper right of the

million square miles — and your job (as commander of the fleet) is to make this whole area into an Exclusion Zone, to stop "your" merchant ships being harassed within it. You have one harbour, with all normal supply facilities, and there are two neutral harbours, which will allow activities such as refuelling and replenishing food stocks, but not re-arming. Not surprisingly, the enemy has a harbour too, with all facilities. The only way you will get to use the enemy harbour for re-supply is to capture it.

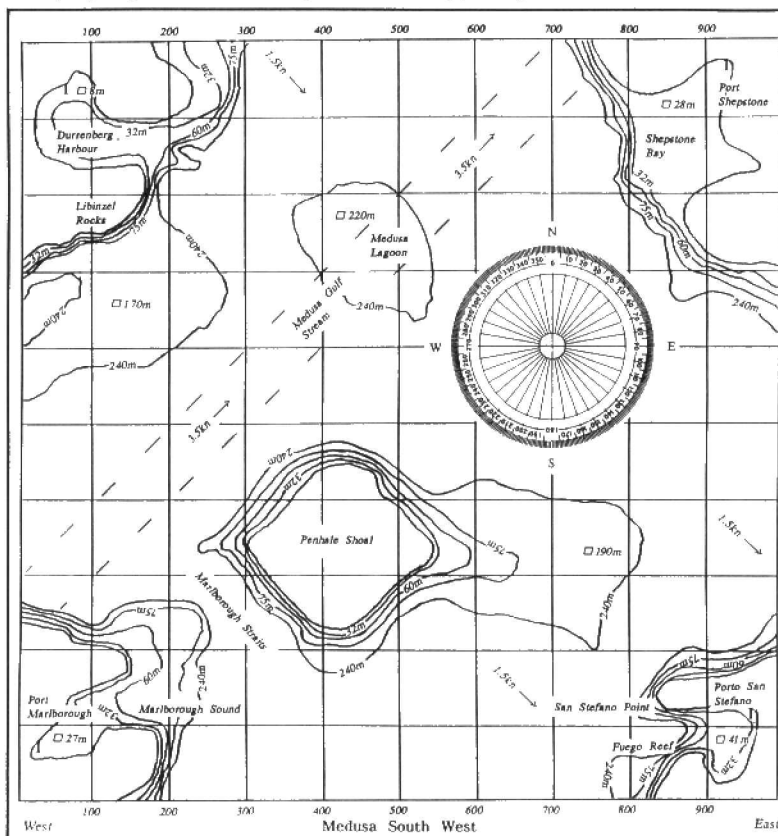
What was unexpected is that the ships are initially in pots which don't have some vital supply facilities, such as fuel. Through not reading all the instructions, I spent a fruitless half hour or so trying to refuel a cruiser at a jetty which had no provision for the job, then proceeded to spend even longer trying (unsuccessfully) to refuel from a stationary tanker in the harbour. As in real life, such operations are difficult: this is no game! In fact, all

ships are fully provisioned at the start of a "scenario", but you can be certain they will be running out of everything by the time you get to grips with the enem, so it is essential to plan ahead for supply stops, and maybe get your fuel tankers moving right from the beginning.

Whatever you can do, your opponent can do too. What's more, unlike chess both parties can move at the same time. You are always likely to be looking for (and aiming at) moving targets. Your ships all star out in one harbour, and you have to manoeuvre them out into the open sea before you can start lookin for the enemy. Getting to sea is no mean task, with 16 ships close together in a smallish harbour, and you are warned that meely getting to the point of the first engagement with the enemy is likely to take about one and a half hours

"real time". This is several days in FTC time: not a lot when you consider that even fast ships only manage about 30-40 miles in an hour, maximum, and average much less normally (or run out of fuel very quickly). Ships don't usually wander arou singly and, while it is much less wearing on your nerves to get them out of harbour one at a time, you have to get them moving in bunches. This means constant scanning of the situation with each ship, in case they are running aground, or into each other.

Even after a few trial runs, my first serious battle attempt read like this after about four hours: six ships aground, two



screen. Suddenly, you are on your own, with only your recollections of the outside situation (plus the internal indicators) to guide you. Sound is used sparingly, and gives the user a good jolt, for example when contact is made with the enemy. Using the Serial linkage causes the sound feature to be inoperative. The displayed time runs at about 35-60 times real time, depending upon how the two computers are connected. That is roughly every second of your time accompanied by the ship's clock stepping on one minute.

Any game has to have a basic scenario, upon which to build your own, more detailed actions. With FTC, you are given an area of 100 x 1000 miles — that is, 1

sunk (one rammed the other — in harbour!), two failed refuelling attempts, and just one contact with the enemy. At that stage of the game, 20 out of 32 ships were (nominally) in motion, and keeping track of them was proving a wearing business. Admittedly, playing yourself is making your job very difficult. One way of getting a rest is to infringe a rule, such as firing weapons in a neutral zone, and get locked-out of the game for a period of several hours (ships' time).

12 hours

In view of the time a battle may occupy — 12 hours of real time is not unlikely — most people will want to save the current scenarios at least once, and restart it some time later. This is easily done, by typing <QUIT> on the command line and pressing ENTER. You then have the option to continue without saving, quit without saving, save and continue, or save and quit. Either QL can be used to initiate the quit process, and the current scenarios will be saved to the default drive on each machine. There is no facility to pause the game, and this makes it necessary to save when you have to answer the door, go to the lavatory, answer the 'phone etc. Failure to halt the game for such interruptions can lead to some nasty crashes. The program will be modified shortly to allow for pausing it.

At first sight, a printer is not essential for playing the game, but its use soon becomes clear. A constant stream of messages appears on the screen, keeping you posted of the numerous incidents occurring on your 16 ships, and there's not much chance of your remembering all of them, or of being able to cope with the demands of each one as it comes. The alternative is to keep checking through all the ships, asking for damage reports wherever progress seems to have been disrupted, but that is a lengthy (and somewhat tedious) business. Similarly, you may think all the bits and pieces which come in the package rather unnecessary, but it does not take long to realise that the charts are essential, and plotting course, making notes of planned dispositions, timing, etc., is the only way to avoid chaos.

Mutiny

If you play it by ear, it seems that the end result is going to be two useless fleets, after a few hours. For instance, moving a handful of ships in each fleet and ignoring the rest results in all manner of problems in those ships are not "allowed to play". Food runs out, the crews become restive, machinery gets sabotaged, and so on. Having tried this approach at first, until all the once-moving ships had run aground after several hours, I found that the problems with the

remaining ships were so numerous that there looked to be little chance of getting any ship moving and the game had to be abandoned in the end.

The entry of commands is an important area with interactive programs. You don't want lengthy typing jobs in the middle of a hectic battle. There are many basic text commands, taking the form of reasonably self-explanatory strings, such as <HARD TO STARBOARD> for putting the ship's rudder fully to the right and initiating a continuous clockwise turn. Such strings would be impossibly long to use in practice and abbreviations are listed — <HARDSTARB> in this case. If a mistake is made in typing, you can retype without deleting the errors. For instance, <HRRHARDSTDSTARB> would be acted upon correctly. The up/down/left/right cursor keys control speed and direction. The CTRL key is used in combination with various others for major commands. CTRL plus up/down arrow give FULL HEAD and FULL ASTERN respectively, and CTRL plus left/right arrow give HARD TO PORT/STARBOARD.

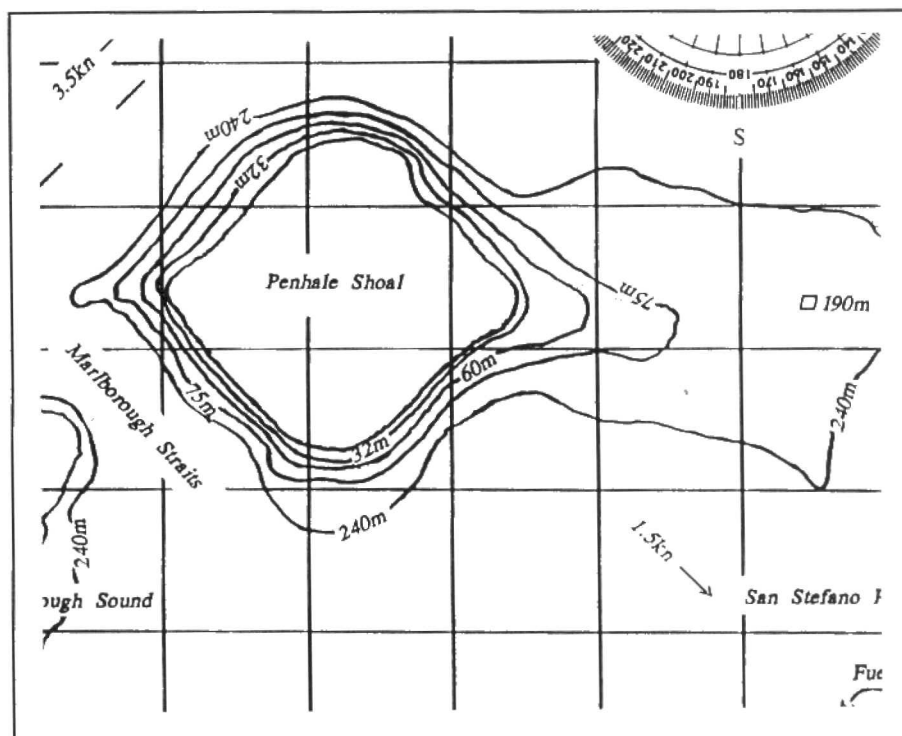
Ship-changing

The familiar job-switching keying of CTRL plus C is reassigned to changing the ship plotted on the IFF (Interrogation-Friend-or-Foe) display. Flooding is dealt with by either <PUMP (area concerned)> or CTRL plus P then <(Area concerned)>. The firing of armaments is handled by the Function keys and the Space bar. The view from the commanded ship is normally directly forwards, but the commands <FRONT>, <REAR>, <LEFT> and <RIGHT> can be used to set the direction.

If TK2 is fitted, the Last Line Recall function can be used. When a command is typed in but not entered for three seconds thereafter, it is removed from the command line.

Some relatively minor problems occurred during review of the program, and the instructions lacked sufficient detail about certain points (while being otherwise very comprehensive). Development is currently continuous and free updates are being sent out. Nothing that happened interfered materially with operation, or significantly impaired enjoyment of the game. A club has been formed, to keep users informed of changes, give advice, and — possibly — as a way of finding playing partners. The conclusion about the program was stated earlier — it is good. It is for the individual to decide whether or not to spend £50 on a "game" (plus many hours of playing time).

Many users, like me, don't go in for games. On the other hands, there are usually (a very few) standard games for every computer, bought by even the most serious users; the case that comes to mind is *Flight Simulator* on the PC, and *Fleet Tactical Command* is comparable with that. To some extent, both games can be treated fairly lightly if you are not ready to study the instructions, but the difference with FTC is that you can't "fly hands off" for long, as is possible with *Flight Simulator*, because things keep happening which require your input. Comparison can also be made with 3-D and postal chess, as the continuously-moving and "multi-compartment" nature of the former is not unlike the problem of managing two fleets of 16 ships each, and the long-distance, stop-and-go nature of the latter should shortly be paralleled by the modem-linked version of FTC.



SOFTWARE FILE

DELUXE FONT ENLARGER

INFORMATION:

Program: DeLuxe Font Enlarger
Price: £19.95
Publisher: Digital Precision, 222 The Avenue, London E4 9SE.

So flexible are the font-handling facilities in Digital Precision's blockbusting *Professional Publisher* program that there are literally billions of combinations of character shape, size, orientation and pattern. The disk on which *Professional Publisher* is supplied is packed with dozens of fonts ranging from the simple to the outlandishly ornate. Yet *Professional Publisher*'s many users have still not been satisfied with the range of larger fonts available to them, hence the arrival of a very clever piece of artificial intelligence in the guise of *DeLuxe Font Enlarger*.

Prisoners

The basic problem with *Pro Pub*'s fonts is that they are prisoners of the pixel matrix (an expression which I confidently predict will one day become the title of a science fiction novel). What appears to be a smooth, curving character is a visual illusion brought about by our minds "adding" information to what our eyes experience. Unfortunately the illusion does not withstand close inspection, and so when the characters are expanded in size the brain refuses to see a continuous curve when it is patently obvious that the "curved line" is in fact a jagged series of rectangular blocks.

Character designs are therefore only completely successful in the scale in which

Gothic text, normal size:

The knights rode in silence to the Monastery of St Asaph where lived their greatest allies - and where their most treacherous enemy was expected

Enlarged within *Pro Pub*:

The knights
St Asaph whe
where their

Enlarged using
Font Enlarger:

The knights
St Asaph whe
where their

Making a larger letter look less like a ladder is a challenge which DP have met, says Mike Lloyd.

they were first designed. They cannot be reduced without immediately losing essential detail and up to now they could not be enlarged except by small pixels into giant blocks. Similarly, while sloping italics are usually quite acceptable (and in *Professional Publisher* are very good indeed), writing text at an angle to the horizontal quickly distorts the shapes of letters to an unacceptable degree.

The advantages of matrix characters are that they are simple to define, take up little storage space in memory and they can very quickly be drawn on the screen. Until the advent of desk-top publishing they were quite acceptable, but almost overnight the drawbacks became cruelly exposed in the headlines of thousands of amateur publications.

Scalable

The alternatives to the prisoners of the pixel matrix are the "scalable font" and "vectorized text", neither term likely to achieve much literary recognition. Scalable fonts are defined in enormous detail but they can be shrunk to any size without losing the essence of their shape. This is done by defining the boundaries of each character, drawing them

at the given scale and filling the inside of the character with the current ink colour. Vectored text does much the same thing but the computer's maths processor is further put to work to calculate each letter's orientation in relation to the page's horizontal. Both techniques required huge quantities of memory space in which to store the font definitions and an extremely powerful computer if text is to be produced at anything like an acceptable speed. It is only with the arrival of laser printers with their own programming language, enormous ram space and powerful cpus that scalable fonts and vectored text have become popular.

Sadly, the QL has yet to see its first scalable font on screen. Stuck with the pixel matrix, the best must be made of it. Programmer Mark Knight clearly felt that not enough was being done and so he programmed his QL to expand and modify existing fonts so that the enlarged characters were as smooth and regular as the originals appeared to be. Digital Precision are now marketing his program, DeLuxe Font Enlarger, at the reasonable price of £19.95.

Expandable

Both QL character sets and Pro Pub high definition fonts (hdfs) can be expanded by the program to any of the legitimate hdf sizes supported by Pro Pub. The artificial intelligence within the program code identified where the pixels are attempting to emulate the solid line or curve and smooths the jaggedness by filling in or taking away pixels. The routines must carefully avoid smoothing angles which are meant to be there, such as in the letters T, E and Z. It must also decide whether serifs (the decorative bits stuck on the extremities of letters) are to be square or shouldered. Sometimes, inevitably, the program will judge wrongly and so a font editor is included so that users can either start a new font from scratch (a particularly long-winded and usually abortive exercise) or modify existing or enlarged fonts. In tests run by *Sinclair QL World* it was found that the program handled the exterior angles of letters extremely well but that concave

curves occasionally unsettled the smoothing logic.

The best results occur when the expansion is limited to double or perhaps treble the original size of the characters. This means that every standard-sized QL font (and over 80 varieties are provided with Lightning alone) can be expanded to the scale of the default "B" HDF font with little or no need for additional editing. The new font can then be imported back into the Font Enlarger and enlarged again if necessary. Alternatively, a huge jump in scale can be achieved in one go provided

for enlargement and loading a font for editing was not made sufficiently clear.

Editing

These niggles are of little consequence, however, in a program which might only be used once per font. Equally, the functionality of the bundled font editor is not terribly important. My preference is for the Pro Pub editing environment, but the important thing is that each user is free to choose to edit characters in his or her preferred way. The seven page manual which comes

constructed, but the results are usually worth the wait and the process need only be done once for each font required.

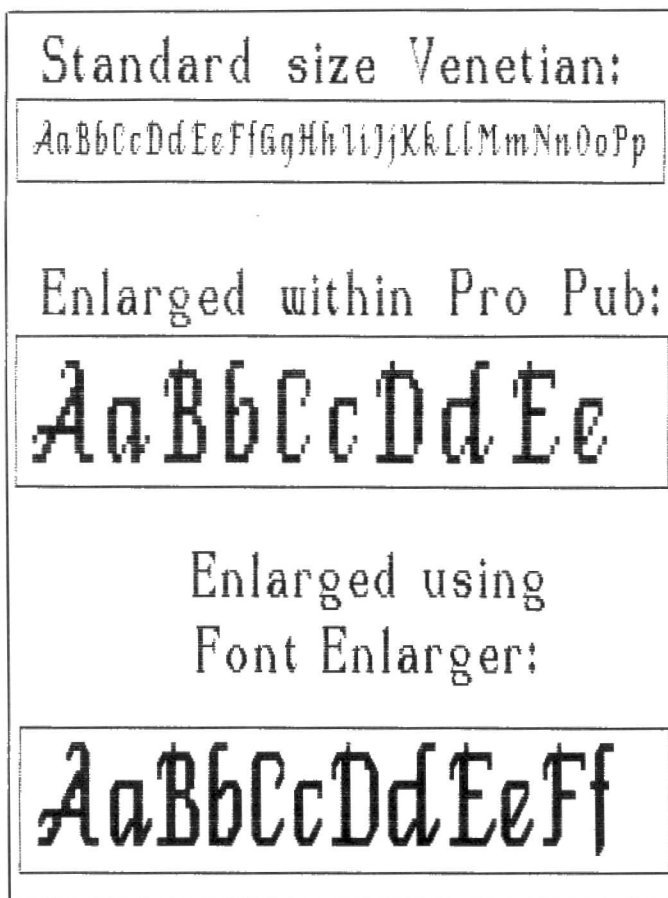
I must confess that the results were initially unsurprising. It is a great compliment to the program that the large typefaces immediately look "right". It was only when the enlarged fonts were compared with the original font expanded using Pro Pub's "photographic" enlargement that the impressive differences between the old and the new became obvious. The typefaces chosen to illustrate this article, a Gothic QLS set and the Venetian HDF font, were selected because of the special character afforded by the shape of their letters. Simple enlargement by doubling the size of pixels quickly destroys the illusion of coherent curves, as is shown by the accompanying illustrations. Could Font Enlarger enlarge these fonts while keeping their special characteristics intact?

Illustrations

The answer is clear to see from the illustrations. The enlarged characters have not been edited in any way, although clearly both styles would benefit from a judicious bit of tidying up. Also, they have both been expanded way beyond their original size: more modest expansion would have resulted in slightly better-formed characters than those shown here. The acid test is to imagine a headline publishing using either the Gothic or the Venetian font: without the benefit of Font Enlarger the results would be totally unacceptable.

Decision

Those who use the QL and Professional publisher for electronic publishing have an important decision to make if they are to remove jagged headlines from their publications. They can consider purchasing an Apple Macintosh or a 386 PC, an upmarket DTP package, a laser printer with Postscript and a set of expensive font cards (and say goodbye to £10,000 in the process) or they can invest in Mark Knight's clever little utility for less than £20. Not a difficult decision to make.



that a little time is invested in tidying up one or two of the characters using the font editor.

The Font Enlarger is not difficult to use because so little needs to be specified by the user beyond a file name containing the source font. Nevertheless, I feel that more attention could have been paid to screen presentation and menu management. It became a little tiresome to be asked time and time again whether the selection I had made was the one I really wanted, and the difference between loading a font

with Font Enlarger is workmanlike if occasionally a little dense.

The value of the Font Enlarger lies not in the way it achieves its goals but in the quality of its output. The time taken to work its way through all the characters of a particular font is not of much importance, but it is worth reporting that a good book is a useful companion while the program is going through its paces. When expanding a QLS font to the maximum size of 4x48 pixels each letter may take thirty seconds or more to be

DIY TOOLKIT

It would be convenient to be able to swap between several programs – or versions of a program – quickly, but the standard QL loads Basic very slowly and limits you to a single program in memory. *MultiBasic* allows SuperBasic programmers to keep several tokenised programs in memory at one time.

MultiBasic can only interpret one program at a given time, but it can swap between programs extremely fast. Swapping may be manual or automatic, controlled by the programs themselves or other tasks. The only limit on the number of programs that can be loaded at once is available memory.

MultiBasic made its debut in the March issue of *QL World*, just before Focus Magazines called in the Receiver. This delayed publication of the promised “fol-

MULTIBASIC PART 2

Simon Goodwin explains the logic behind his DIY Toolkit MultiBASIC project.

low up” article, but I shall now explain how MultiBasic works and how programmers can add extra features. Technical terms were explained in past columns, particularly in the August 1988 issue. I have summarised the advantages of MultiBasic in the Panel.

Command Summary

UNLOAD makes a named copy of the current SuperBasic system in task mem-

ory. The original copy of the program remains in SuperBASIC until you delete it or use RELOAD.

If the name you UNLOAD is in use MultiBasic reports ‘already exists’. MultiBasic will not create two tasks with the same name. RESAVE is like UNLOAD, but replaces any existing task with the same name.

RELOAD copies a stored program from task memory to SuperBasic. It expects the same name you used with UNLOAD or RESAVE. MultiBasic programs share the same format and area of memory with other tasks, so you can manipulate them with standard Toolkit commands.

Common extensions like JOBS or LIST_TASKS will reveal the MultiBasic tasks. Likewise you can dispose of those tasks with RJOB, REMOVE TASK or *DIY Toolkit’s* PURGE command.

MultiBasic’s own REMOVE command can erase any task, given its name in upper or lower case. If you load the same machine code task several times REMOVE will get rid of the first one with a matching name, every time you use it. REMOVE scans tasks in the same order as JOBS and LIST TASKS.

MultiBasic reports ‘out of memory’ if there’s not enough free memory to UNLOAD or RESAVE a task. It complains ‘bad parameter’ if you do not specify a valid task name. It accepts any sequence of 1–127 characters, in single or double quotes – or a string variable or expression.

Unquoted names are allowed, as long as they are not used for some other purpose in the program. The file commands SAVE, LOAD and LBYTES use the same rules, although they do not allow such long names. The shortest practical names are single letters or digits.

The ‘invalid job’ reports means that you’ve tried to RELOAD or REMOVE a task that does not exist, or there are so many tasks that QDOS has not got room to keep track of another one (which is very unlikely). You may get a ‘channel not open’ error if you RELOAD a task that was using channel numbers above #2 when you UNLOADED it.

MultiBasic is listed in two forms. **Listing One** is the commented assembly code. You need an assembler to convert this

Using MultiBasic

MultiBasic is specially useful because BASIC loading is so slow on the QL. The QL reads program text in 512 byte lumps which it converts into internal “tokens”. Disk loading is almost as slow as microdrive; even ram disks are painfully slow, because the conversion into tokens takes much longer than fetching the raw data.

MultiBasic stores programs in the tokenised form, so they can be moved directly into the interpreter’s memory. MultiBasic moves from 270K to a megabyte per second, depending on the speed of your system.

SuperBasic loading gets slower and slower as program size increases, but MultiBasic’s RELOAD time is proportional to the length of your program, and under a second even for very long programs. Once you’ve loaded a program you need never re-load it the slow way unless you run out of memory or turn off the computer. This is great when working with Basic compilers, which expect you to load programs into the interpreter before you compile them.

MultiBasic makes it easier to write often-used utility programs in Basic, as there is no need to re-load (and tokenise) them whenever they are needed. It is particularly useful for ‘template’ programs that require regular tweaks for specific tasks. It speeds up SuperBasic development and testing. You can UNLOAD a copy of a program before making changes, and RELOAD instantly if problems crop up in the revised version. It’s still a good idea to SAVE to disk or cartridge every

so often, in case you lose the entire contents of memory. Thankfully, SAVE is much faster than LOAD.

The complete programming environment is stored by MultiBasic, including variable values, arrays, and temporary results. You can UNLOAD a program at any time – even while it is running – and when you RELOAD it, it will carry on where it left off. MultiBasic remembers local variables and procedure calls, so you can break into a program, UNLOAD it, and CONTINUE later. In the meantime you could load or RELOAD any number of other Basic programs.

It is quite acceptable to use UNLOAD inside a running program. When the program is RELOADED it will continue running from the point after the UNLOAD statement, as if it had never stopped. You can arrange sequences of programs which load one another and communicate through the MEM device, featured in *DIY Toolkit* last summer.

When you UNLOAD or RESAVE a MultiBasic task, the current command line is saved along with the program. Any commands after the UNLOAD will be executed as soon as you RELOAD that image, so UNLOAD prog:LIST saves ‘prog’ in such a way that it is LISTed automatically whenever you RELOAD it later.

Alternatively, you can make a program RUN as soon as it is RELOADED by saving it like this: RESAVE prog:RUN. Statements after RELOAD are ignored; like LOAD and LRUN, RELOAD discards the remainder of the command-line when it loads a new program.

LISTING ONE

* QL WORLD DIY TOOLKIT - dynamic BASIC swapping routines
* Version 3.3 Copyright 1988-90 Simon N Goodwin

```
*
looper equ 24830      Opcode of BRA.S *-2
*
start lea.l define,a1
      move.w $110,a2      BP.INIT vector
      jmp (a2)           Add new commands
*
*   SHARED KEYWORD CODE
*
unload moveq #-1,d7      Flag for UNLOAD
      bra.s getname
reload moveq #1,d7       Flag for RELOAD
      bra.s getname
resave moveq #-2,d7      Flag for RESAVE
      bra.s getname
remove moveq #0,d7       Flag for REMOVE
*
getname lea 8(a3),a4      Check for one parameter
      cmp.l a4,a5
      beq.s one_par
bad_par moveq #-15,d0     Report BAD PARAMETER
dropout rts
one_par tst.b 0(a3,a6.l)  Does it have a value?
      beq.s unset        No, use the name instead
      move.w $116,a0      CA.GTSTR vector
      jsr (a0)           Get the value as a string
      bne.s dropout      Give up if that went wrong
      move.l $58(a6.l),a1  Get string offset (BV.RIP)
      tst.b 0(a1,a6.l)    Check length is 0-255
      bne.s bad_par      No good, length >= 256
      lea.l 1(a1),a5      0(A5,A6.L) -> length byte
      bra.s tst_len
*
unset move.l 24(a6),a0     Find BASIC's Name Table
      move.w 2(a3,a6.l),d0  Index of actual parameter
      lsl.w #3,d0         N.T. entries take 8 bytes
      adda.w d0,a0        (A0,A6.L) -> N.T. Entry
      move.w 2(a0,a6.l),d0  DO is name offset in list
      move.l 32(a6),a5     Find BASIC's Name List
      add.w d0,a5          0(A5,A6.L) -> length byte
tst_len tst.b 0(a5,a6.l)  ** 3.3 ** Allow 1-127 chars
      ble.s bad_par
*
* Look up the required name amongst all the tasks
*
look_up moveq #0,d1        Start with task 0,0
      bra.s scan          Scan all the tasks
look_on tst.l d1           At end of task tree?
      beq.s do_cmd        If so, do the command
scan moveq #0,d2           Scan from top of tree
      move.l d1,d4        Save current task ID
      moveq #2,d0         MT.JINF trap key
      trap #1             Find the next task
      tst.l d0            Does it exist?
      bne.s do_cmd        If not, do the command
*
* A0 -> task, D4 is task ID, D1 is ID of 'next' task
*
```

```
move.l a0,d6             Save base address of task
addq.l #2,a0             Skip BRA.S at the start
move.l (a0)+,d5          Get length
cmpl.w #$4AFB,(a0)+     Does task have a header?
bne.s look_on           No, don't look for a name
tst.b (a0)+             Name length should be <256
bne.s look_on           Otherwise, keep looking
move.b 0(a5,a6.l),d0     Get parameter name length
cmp.b (a0)+,d0          Do name lengths match?
bne.s look_on           No, try another task
move.l a5,a4            Copy name pointer for test
compare move.b 1(a4,a6.l),d2  Get parameter character
moveq #-33,d3           Mask to ignore letter case
and.b d3,d2             Convert parameter to caps
and.b (a0)+,d3          Convert header to capitals
cmp.b d2,d3             Compare characters
bne.s look_on           Look on unless they match
addq.l #1,a4            Advance through parameter
subq.b #1,d0            Count one less to check
bne.s compare           Check all the characters
moveq #-1,d1            Flag that name was found
*
* If D1 is negative, task was found: code at D6, length D5
*
do_cmd tst.l d7          What operation is needed?
      bmi.s unloda        Negative, UNLOAD / RESAVE
      bne reloda         Positive, RELOAD
*
* REMOVE routine - if we found a task, we can remove it
*
remove tst.l d1          Was the name found?
      bmi.s kill_it      Report INVALID JOB
bad_job moveq #-2,d0
      rts
exists moveq #-8,d0      Report ALREADY EXISTS
      rts
unknown moveq #-7,d0     Report NOT FOUND
      rts
*
kill_it move.l d4,d1      Set up task ID
      moveq #5,d0        MT.FRJOB trap key
      trap #1            Remove the task
      rts               Report back to caller
*
* UNLOAD routine - copies BASIC to an image in task RAM
*
unloda tst.l d1          Did we find the name?
      bpl.s make_it      No, so create it
      addq.w #1,d7        Is this UNLOAD?
      beq.s exists       If so, it already exists
      bsr.s kill_it      Otherwise it's RESAVE...
make_it moveq #0,d0      MT.INF trap key
      trap #1            Find system variables
      trap #0            Stop anything moving!
      or.w #$700,sr      Disable interrupts
      move.l a0,a3        A3 -> system variables
      move.l 20(a0),d3    D3 -> End of BASIC+1
      sub.l 16(a0),d3     D3 = Length of BASIC area
*
* D3 is length of 'data' area. Get 'code' length into D2.
*
```


LISTING ONE continued

```

moveq #18,d2      D2 = Minimum 'code' header
move.b 0(a5,a6.l),d5 Find length of task name
addq.b #1,d5      ... and ensure it is even
bclr #0,d5        Round length to even
add.b d5,d2       Add name to 'code' length
*
* Allocate transient program memory for the image
*
alloc8 moveq #0,d1      Owner: SuperBASIC
      move.l d1,a1      Put task in TRNSP
      moveq #1,d0       MT.CJOB trap key
      trap #1           Create a transient task
      tst.l d0          Check for errors
      bne.s unlock      Give up if CJOB failed
*
* Set up a dummy task header
*
      move.w #looper,(a0)+ START is BRA.S START !
      move.l d3,(a0)+    Store length AFTER header
      move.w #$4AFB,(a0)+
      add.l a6,a5        A6 cannot change now
      move.b (a5)+,d0     Get length of task name
      ext.w d0
      move.w d0,(a0)+    Store length (always >0)
setname move.b (a5)+,(a0)+ Copy name a byte at a time
      subq.b #1,d5       Use even count in D5 to
      bne.s setname      ... ensure A0 ends up even
      move.l a6,(a0)+    Save base address of BASIC
      move.l usp,a5      Save User Stack Pointer
      move.l a5,(a0)+
*
      move.l 16(a3),a3    A3 -> NEW start of BASIC
copy_d3 lsr.l #3,d3      Count pairs of long words
      subq.l #1,d3       Adjust for DBRA loop
copier  move.l (a3)+,(a0)+ It's fast and safe to move
      move.l (a3)+,(a0)+ ... eight bytes at a time
      dbra d3,copier     Move 270-540 QL K a second
lockout moveq #0,d0      Report no error
unlock  andi.w #$D8FF,sr Re-start QDOS
      rts
*
* RELOAD <name> - fetch task from (D6), length in D5
*
reloda  tst.l d1         Did we find the name?
      bpl.s unknown
load_it move.b 0(a5,a6.l),d7 Find name length for later
      addq.b #1,d7

```

```

bclr #0,d7      Round length to even
moveq #0,d0     MT.INF Trap key
trap #1         Find system variables
move.l a0,a4     Save system var. pointer
move.l 20(a0),d1 D1 -> End of BASIC+1
sub.l 16(a0),d1  D1 := Length of BASIC area
sub.l d5,d1      D1 is delta size
beq.s size_ok    No change required
bpl.s shrink     Discard excess space
neg.l d1         Expand by D5-D1 bytes
moveq #22,d0     MT.ALBAS trap key
trap #1         Expand BASIC work space
tst.l d0         Check for OUT OF MEMORY
beq.s size_ok
rts
shrink moveq #23,d0 MT.REBAS trap key
trap #1         Shrink BASIC work space
*
* Now work out the address of the code in the task space
*
size_ok moveq #10,d2 Skip fixed part of header
      add.b d7,d2      Skip name (max length 127)
      add.l d2,d6      Point D6 at saved A6 & A7
      move.l d6,a3     Set source pointer
      trap #0          Stop BASIC space moving
      or.w #$700,sr    Disable interrupts
*
      move.l a6,a0      Offset user A7 by delta A6
      sub.l (a3)+,a0     Delta address of BASIC
      add.l (a3)+,a0
      move.l a0,usp     USP is delta A6 + old USP
      move.l 16(a4),a0  Get destination, SV.BASIC
      lsr.l #3,d5       Count long word pairs
      subq.l #1,d5      Adjust for DBRA
copy  move.l (a3)+,(a0)+ Move task data to BASIC
      move.l (a3)+,(a0)+
      dbra d5,copy
      bra.s lockout     Restart QDOS and exit
*
define dc.w 4      Number of procedures
      dc.w unload-*
      dc.b 6,'UNLOAD'
      dc.w reload-*
      dc.b 6,'RELOAD'
      dc.w resave-*
      dc.b 6,'RESAVE'
      dc.w remove-*
      dc.b 6,'REMOVE'
      dc.w 0,0,0     No functions
end

```

into code you can run; I use Hisoft *Devpac*.

Listing two is a simple Basic loader which reads the machine code from data statements and stores it in a file. It's easier to enter this listing, especially as lines 100-580 are the same for every *DIY Toolkit* package, but it is relatively hard to customise the code. Once you have created the code file this will add the new commands to SuperBasic:

base=RESPR(446)

LBYTES "filename",base
CALL base

Thereafter you can use UNLOAD, RELOAD, RESAVE and REMOVE in your own programming.

MultiBasic has been tested on the Thor XVI, QDOS (including *Minerva*, 1.63) and the Public Domain QL emulator for the Commandore Amiga. It is reliable and easy to use, but there are a few traps for the unwary. The next three sections explore what happens if you load keywords, leave channels open or run out

of memory while using it.

MultiBasic keeps additional SuperBasic programs in the QDOS task area. They are less likely to cause memory fragmentation than if they were allocated in the common heap, where they would be mixed up with device and channel details stored as programs run.

Task memory may still become fragmented. You can end up unable to store a large task because the available memory is split into smaller chunks. QDOS task must be held in a single continuous area,

but QDOS does not allow tasks other than SuperBasic to move after space is created for them in memory.

In theory MultiBasic tasks are relocatable without change. They hold moveable SuperBasic data-structures and automatically re-calculate the value of registers A6 and A7 after loading. Sadly there is no way to move tasks with documented QDOS calls; in any case fragmentation might be a problem if executable tasks get mixed with Basic images.

Tasks share one area of memory, at addresses higher than those used by the current SuperBasic program. QDOS stores tasks in fast memory whenever that is available. New tasks are loaded to the bottom (low address) of the task area, moving the current SuperBasic area down to make room, unless there is unused space for the entire new task higher in task space. Such gaps crop up when a previously-running task is removed, and another task or image has been loaded in the meantime.

SPACE

Imagine if you were to UNLOAD a Basic program, then UNLOAD a second, RELOAD the first and add a few lines, then try to RESAVE it. You may find that the re-saved program does not fit back into the space it originally occupied. You end up with an unused space, large enough for the original program, followed by the second task. The expanded copy of the first task will be stored the other side of the second one.

If you must avoid fragmentation of memory, always REMOVE tasks and images in reverse order to the sequence in which you loaded them. Last in should be first out. In practice fragmentation is unlikely to be a problem unless you are short of memory.

When an image is copied the original is left in the BASIC area. Similarly, RELOAD does not remove the stored task image, so there must be enough free memory for two copies of any task that is to be UNLOADED or RELOADED.

SuperBasic programs tend to grow as they run. Variables and calls to definitions cause dynamic areas to expand. CLEAR and NEW empty some of the area, but they do not reset them all, so the SuperBasic environment tends to get bigger.

If you have the SYSBASE function from the December *QL World* you can display the size of the SuperBasic area. This includes tokens, variables and tables like the Name Table and Name List, which record the names and addresses of resident procedures and functions. This prints the size of the SuperBasic area, in bytes:

```
PRINT PEEK_L (SYSBASE+20)-
PEEK_L (SYSBASE+16)
```

LISTING TWO

```
100 REMark Sinclair QL World HEX LOADER
110 REMark by Marcus Jeffery & Simon N Goodwin
120 :
150 CLS: RESTORE : READ space: start=RESPR(space)
160 PRINT "Loading Hex..." : HEX_LOAD start
170 INPUT "Save to file...":f$
180 SBYTES f$,start,byte : STOP
190 :
200 DEFine FuNction DECIMAL(x)
210 RETurn CODE(h$(x))-48-7*(h$(x)>"9")
220 END DEFine DECIMAL
230 :
240 DEFine PROCedure HEX_LOAD(start)
290 byte = 0 : checksum = 0
300 REPEAT load_hex_digits
310 READ h$
320 IF h$="*" : EXIT load_hex_digits
330 IF LEN(h$) MOD 2
340 PRINT"Odd number of hex digits in: ";h$
350 STOP
360 END IF
370 FOR b = 1 TO LEN(h$) STEP 2
380 hb = DECIMAL(b) : lb = DECIMAL(b+1)
390 IF hb<0 OR hb>15 OR lb<0 OR lb>15
400 PRINT"Illegal hex digit in: ";h$ : STOP
420 END IF
430 POKE start+byte,16*hb+lb
440 checksum = checksum + 16*hb + lb
450 byte = byte + 1
460 END FOR b
470 END REPEAT load_hex_digits
480 READ check
490 IF check <> checksum
500 PRINT"Checksum incorrect. Recheck data.":STOP
520 END IF
530 PRINT"Checksum correct, data entered at: ";start
560 END DEFine HEX_LOAD
570 :
580 REMark Space requirements for the machine code
590 DATA 446
600 :
610 REMark Machine code data
620 DATA '43FA018C34790000','01104ED27EFF600A'
630 DATA '7E0160067EFE6002','7E0049E0008BBCC'
640 DATA '670470F14E754A33','E800671A30790000'
650 DATA '01164E9066EE226E','00584A31E80066E2'
660 DATA '4BE900016016206E','00183033E802E748'
670 DATA 'D0C03030E8022A6E','0020DAC04A35E800'
680 DATA '6FC0720060044A81','673C740028017002'
690 DATA '4E414A8066302C08','54882A180C584AFB'
700 DATA '66E44A1866E01035','E800B01866D8284D'
710 DATA '1434E80176DFC403','C618B60266C8528C'
720 DATA '530066EC72FF4A87','6B1C660000884A81'
730 DATA '6B0C70FE4E7570F8','4E7570F94E752204'
740 DATA '70054E414E754A81','6A06524767E861EE'
750 DATA '70004E414E40007C','0700264826280014'
760 DATA '96A8001074121A35','E800520508850000'
770 DATA 'D405720022417001','4E414A80663030FC'
780 DATA '60FE20C330FC4AFB','DBCE101D488030C0'
790 DATA '10DD530566FA20CE','4E6D20CD266B0010'
800 DATA 'E68B538320DB20DB','51CBFFFA7000027C'
810 DATA 'D8FF4E754A816A82','1E35E80052070887'
820 DATA '000070004E412848','2228001492A80010'
830 DATA '928567126A0C4481','70164E414A806706'
840 DATA '4E7570174E41740A','D407DC8226464E40'
850 DATA '007C0700204E91DB','D1DB4E80206C0010'
860 DATA 'E68D538520DB20DB','51CDFFFA609E0004'
870 DATA 'FE7C06554E4C4F41','4400FE780652454C'
880 DATA '4F414400FE700652','455341564500FE6A'
890 DATA '0652454D4F564500','000000000000','*',40335
```

Minerva users have the option of calling VER\$(1-2) instead of SYSBASE; Thor users can substitute SYS_VARS.

The code to expand the SuperBasic area is rather slow. To minimise the time spent grabbing space, QDOS always allocates at least 512 bytes extra whenever it runs out of room. This granularity means that a few hundred bytes may be allocated but initially unused. However it increases the chance that MultiBasic will be able to re-use the old space when you RESAVE after adding a few variables or statements.

Toolkit's CLOSE, channels with SuperBasic numbers greater than 2 may need to be re-opened when you RELOAD.

If you type NEW, LRUN or LOAD between the UNLOAD and RELOAD commands the system will close all channels except the standard windows #0, #1 and #2. When the program you RELOAD attempts to use those channels' BASIC reports CHANNEL NOT OPEN.

The easy way to avoid this program is not to UNLOAD with extra channels open. Another good way to avoid trouble is the UNLOAD an empty image at the start of

commands that clear variables, such as CLEAR, NEW, LOAD and LRUN, also release memory allocated with the Super-Toolkit 2 ALCHP function. Avoid using these commands while a task that called ALCHP is waiting to be RELOADED.

Compiler users may find it useful to UNLOAD an image containing only those toolkit commands they actually use in programs, before loading other toolkits. The fewer commands an image knows about, the faster SuperBasic loading will be and the smaller the compiled task. Each resident command defined at compile-time bumps up the size of the resultant task by 4 bytes (*Supercharge* or *Turbo*) or 8 Bytes (*Qliberator*), even if you do not use the command in your program.

The program

The MultiBasic codes implements four commands, UNLOAD, RELOAD, RESAVE and REMOVE. The listing runs to less than 300 lines of assembler, and 400-odd bytes of code, but it does a lot of work. This section explains the assembly codes, so you can read and customise it as you wish.

The program was originally assembled with HiSoft's Devpac, but this listing has been modified to accommodate crude assemblers that do not recognise 'generic' instructions, following comments from M Cadman, who uses Metacomco's slow and pedantic *QL Assembler Development Kit*. The font has also been changed, in a bid to make characters like 'B' and '8' easier to distinguish.

The code starts in the usual way, linking the tables of new commands labelled DEFINE into the SuperBasic interpreter. All four commands expect a single parameter — the name of a task — so they start by executing the same code. When the parameter has been fetched the value in register D7 identifies the particular action to be performed.

The code of GETNAME checks that only one task name has been supplied, and reports 'bad parameter' otherwise. It would be annoying to have to type each task name in quotation marks or inverted commas, so MultiBasic has two different ways of finding the task name.

If the parameter has been assigned a value the code users CA.GTSTR to read the value as text. It checks that the high byte of the length of the result is zero, excluding names more than 256 bytes long! If all goes well A5 ends up pointing to a byte showing the length of the text, followed by the text itself. This code is used if you specify a variable name or expression as a parameter. It allows UNLOAD "name", RELOAD name\$ & 2 or RESAVE 2, for example.

If the parameter has no value the code labelled UNSET tracks down the 'actual parameter' name used in the command. The second word among the parameter details at (A3,A6) is the Name Table Index of the parameter name. The code shifts this to allow eight bytes for each entry in

Figure 1. — Map of a SuperBasic program in task memory.

| | | | |
|------|------------------------------|-------|---------------------------------------------------------------------------------------------------------------------------------------------------------|
| | start | bra.s | start |
| TASK | | dc.1 | prog__length |
| | | dc.w | \$4AFB |
| CODE | | dc.w | name__length |
| | | dc.b | 'Name text' |
| AREA | | ds.w | 0 |
| | | dc.1 | Saved__A6 |
| | | dc.1 | Saved__A7 |
| | prog prog+104 prog+360 | | |
| TASK | | | BASIC Task header BASIC system variables BASIC INPUT/COPY Buffer BASIC Token List BASIC Program File BASIC Name Table BASIC Name List |
| DATA | | | BASIC Variable Values BASIC Channel Details BASIC Return Table |
| AREA | | | BASIC LIST Line Number Table BASIC Backtrack Stack BASIC Temporary Graph Stack BASIC Arithmetic Stack BASIC System (USP) Stack |

Figure 1 shows the internal contents of a task that has been UNLOADED. The 'code area' is at least 18 bytes long, and holds the task name and other details that are not part of the Basic area. The 'data area' holds an exact copy of the SuperBasic area at the time you UNLOADED. This starts with 104 bytes of task header — the 'Job Control Block' for SuperBasic — followed by 256 bytes of Basic system variables, then the twelve dynamic tables and stacks used by the interpreter.

Only the first word of the code area is executable. It's an infinite loop, and serves to stop the task crashing if it somehow becomes activated. This should not happen, but if it does the task just runs round and round in a tight loop till it is time for another task to run. There is no executable code in the SuperBasic area, so it cannot run independently of the interpreter in the ROM.

Channels

If you leave channels open and UNLOAD a task, then reload it after issuing commands like NEW or Super-

your programming session: NEW:UNLOAD Empty. Thereafter, when you would normally type NEW, enter: RELOAD Empty.

This does not close channels, and often releases more memory than the standard NEW command. If you forget to save an empty program and need to delete a program with using NEW, try DLINE 1 to 32767: CLEAR; this discards variable and program lines, without closing channels. Use MERGE and MRUN to load another program.

If you load SuperBasic extensions after UNLOADing a program, those extensions will not be available when you RELOAD the image, because it only includes names defined at the point when you UNLOADED. Missing procedures cause a 'bad name' error, while absent functions give 'error in expression'.

To retain access to those commands you must UNLOAD an image which does recognise those names before RELOADing the old image. To retrieve the commands, RELOAD the newer image.

SuperToolkit users should note that

the table, then extracts the offset of the name text in the Name List. Again it winds up with A5 pointing at the length, in one byte, followed by the text.

For instances you might enter UNLOAD FRED. This creates a new, value-less entry in the SuperBasic Name List for the name FRED, as long as you are not using that name for some other purpose.

If you choose a name that is already in use the code will attempt to find its value and use that. This provokes an expression error if you use a procedure name, and 'bad line' if you use the name of a keyword like REM or DEF. Otherwise the result depends on the value of the name:

UNLOAD PI

happily creates a task called '3.141593'. Of course the same can happen with standard commands that allow names, like LRUN and MERGE. You can use any name, as long as you put it in quotation marks so that MultiBasic knows that it should take you literally:

UNLOAD 'PI'

The code at TST_LEN signals a parameter error unless the name consists of between 1 and 127 characters. I have changed this line slightly, after Bill Lawrence reported problems UNLOADing large programs. The new version uses explicit long addressing for A5 and A6, rather than leave the addressing mode to the whim of the assembler.

To change the hex version listed in March, alter line 670 to end "35E800", instead of "36D000", and alter the checksum on line 890 from 40312 to 40315. Alternatively, load the old code to address X and POKE X+93,53:POKE X+94,232 before you use MultiBasic. This tweak prevents spurious 'bad parameter' reports if you UNLOAD big listings.

Once the required task name is known the next step is to look through the task to see if that name is already in use. We need this information whether we are reloading, removing or unloading a task. MT.JINF tells the start address of each task in turn. MultiBasic tasks start with a standard sequence of data, shown in **Figure 1**. The first word is a loop instruction, which soaks up processor time harmlessly if the task somehow becomes active; MultiBasic tasks are normally dormant as they contain tables of data for the SuperBasic interpreter, rather than executable code.

The next long word indicates the total size of the task. It is followed by the task name, in the standard format expected by commands like JOBS and LIST_TASKS. The 'illegal' instruction \$4AFD marks the start of a data area, followed by the length of the task name (a word) and the name text.

If the name length matches, the loop labelled COMPARE checks each charac-

ter against the parameter. It would be annoying if capitals and small letters had to match exactly. File commands tread 'lower' and 'UPPER' case letters as equivalent, so it would be consistent if REMOVE FRED could match 'Fred' or 'fred', rather than give an error report.

The QL rom includes a vector, UT.CSTR, to compare strings of characters. This can ignore case, but it's hard to use because it requires that both strings start with a length word at an even address, offset from register A6.

The Name List is easily accessible via A6, but the length is just a byte, often at an odd address. The text in the task header starts with a length word, at an even address, but it is outside the SuperBasic area so it does not keep track if A6 changes!

So we need our own way of obscuring the difference between capitals and little letters. It must be simple, reasonably fast, and should take account of the accented capitals with codes from 160 to 171, and the corresponding small letters from 128 to 139. Other capital use codes from 65 to 90, while lower case runs from 97 to 122.

We might use a conversion table of 256 codes, but that's rather inelegant, and I would hate you to have to type in something so dull. There must be a better way.

The difference between the codes of capitals and lower case is always 32. It follows that you can make capitals and lower case equivalent (among other things) if you clear bit 5 of the binary pattern of each character before making comparison.

Bit 5 is the bit that carries the value 32, as 2 to the power of 5 is 32. The value -33 in the listing is a signed representation of the binary byte %11011111 - a 'mask' obscuring bit 5. I use decimal this month, to avoid confusing Metacomco.

The snag of this conversion is that it affects some characters which are not letters. The copyright sign, for instance, has Ascii code 127; it matches an underscore, code 95, if you strip out bit 5.

In practice this hardly matters. The digits and punctuation codes from 32 to 63 only give a false match for characters 0-31. Apart from Chr\$(10) - the new line character - those codes cannot be entered in SuperBasic anyway.

The scheme means that square and curly brackets are considered equivalent, but that is no great problem. The At and Pound signs match, as do backslash and vertical bar. The only other ambiguous pair in the Ascii sequence 0-127 is the caret, which matches a tilde character.

The loop at COMPARE tweaks the parameter every time it checks a task name. It would often be more efficient to capitalise the characters of the parameter once and for all, but that would involve more code and space is tight.

By the time the 68008 reaches the label DO_CMB register D1 has a positive value if we have found the task specified

by the parameter; otherwise D1 is negative.

D7 indicates the action to be performed; it is positive after UNLOAD, and negative if we're loading. D7 is zero if the task must be removed. REMOVA starts by checking D1, and reports 'invalid job' if the chosen name was not found. Otherwise it disposes of the task by calling the system routine MT.FRJOB.

The code for UNLOAD and RESAVE is almost identical. The only difference is the UNLOAD reports 'already exists' if the task name is in use, whereas RESAVE deletes the old task, if any, before saving the new one.

Code labelled MAKE_IT creates a new task holding a copy of the SuperBasic area. Interruptions are disabled while the copy is made; things might get lost if Basic moved part-way through the copying process.

MT.INF finds the system variables that point at the start and end of Basic. The difference, in D3, becomes the size of the task 'data' area. The next block of code works out the size of the 'code' area, which holds the task name, etc.

In this case the terms 'code' and 'data' are not strictly accurate, as both contain data, but Qdos allows task memory to be allocated in two chunks, so we might as well use them both. The space is allocated by calling MT.CJOB. If that succeeds we set up the code area to match **Figure One**.

The last two long words hold the value of A6, The base address of Basic, and the User Stack Pointer. This is normally register A7, but the 68008 has a distinct A7 used by supervisory routines; USP returns the address of the User Stack when the Supervisor stack is selected.

The contents of USP and A6 change as memory is allocated or deallocated. They must be stored along with the task so MultiBasic can work out the correct values when it reloads a program.

The loop at COPY_D3 copies the contents of the SuperBasic area into the task space. It copies eight bytes at the time, for speed; like most Qdos areas, SuperBasic is always a multiple of eight bytes long. The code at UNLOCK restarts multi-tasking.

RELOAD labels the routine to reload a task. If register D1 is positive the specified task was absent, and we give up with a 'not found' report. Otherwise we need to adjust the size of the SuperBasic area to accommodate the store task, restore the saved task image, and set A6 and A7 to point into the SuperBasic area as before.

There are two system calls to change the size of the SuperBasic area. MT.AL-BAS expands Basic by D1 bytes, and MT.Rebas makes it smaller. No toadstool is required. The code after LOAD IT works out the change and passes it to the appropriate routine as necessary. It's a pity you can't make do with one call, and a signed parameter. It's also a shame that you can't say where in the SuperBasic

area you need space. The routines always make room between the LIST line number table (BV.LNP) and the parser backtrack stack (BV.BTP). It's likely that you'll need the space somewhere else and have to move data again, and adjust loads of pointers, to make room.

The Spectrum rom has two entry-points labelled RECLAIM and MAKE ROOM. These let you specify the start and size of the relevant space, and automatically adjust pointers accordingly. It's a pity SuperBasic doesn't.

MultiBasic is only concerned with the total size of the SuperBasic area; it fetches all the necessary data from task memory. The tricky bit is to make sure that A6 and A7 point in the correct places.

Tasks may have been loaded, removed or replaced since the values of A6 and A7 were stored. The current SuperBasic program 'floats' at lower addresses than tasks, so changes in the task area necessitate changes to A6 and A7.

MultiBasic uses the resultant value of A6 after the Basic area has expanded or contracted to fit the new task. The new value of A7 is calculated from the old value, allowing for the change in of A6.

This version of MultiBasic is but a taste of a larger utility which I hope to publish. In the meantime you can enhance MultiBasic by adding extra routines.

It would sometimes be convenient if MultiBasic were to save the screen dis-

play, as well as the SuperBasic area, when you unload or RESAVE a program. This makes MultiBasic seem much more impressive, as it puts you right back where you were, with the same windows on display, as soon as you RELOAD. However you need an extra 32K to save and restore a full QL screen.

Utilities like *Taskmaster* and *QRAM* are not disturbed by MultiBasic, but they do not save and restore MultiBasic displays. They can do this for tasks, by intercepting attempts to open and close display channels, but they don't notice when a SuperBasic program changes, as MultiBasic doesn't need to open or close channels.

In principle it is easy to add screen-saving to MultiBasic. You just allocate an extra 32K in the task area as you UNLOAD, and copy the entire display into the space. When you RELOAD you copy back the old display from the task.

In practice there are complications. You may have changed display mode since you type UNLOAD, so the RELOAD routine should check the current mode and the old one, and if necessary change it before restoring the screen. The system variable SV.MCSTA (PEEK(SYBASE + 52), or Supervisor A6 offset \$34.B) holds 8 if the system is in MODE 8, zero for MODE 4, or 12 for the Thor MODE 12.

Ideally you should also save and restore colours, cursor and boundary positions of the standard windows #0, #1 and

#2, used for commands, program output and listings. You get at these with EXTOP, a favourite *DIY Toolkit* system call. You can make Toolkit 2's ALT ENTER recall lines typed just before the UNLOAD command, by saving and restoring the contents keyboard buffer.

It can be useful to be able to swap two Basic tasks, so that one is RESAVED as the other is RELOADED. This is most convenient if you implement parameter passing between the tasks. Parameters are best held outside the tasks.

You can record tasks on cartridge or disk by saving the task memory with LBYTES. Loading with LBYTES is dramatically quicker than LOAD, but there's a potential snag.

Before you replace the SuperBasic area on RELOADing you must ensure that the address of command routines in the new Name Table match corresponding names in the current SuperBasic area. Otherwise your saved code will call the wrong places when trying to use resident procedures and functions, and crash.

Resident procedure code is likely to be a different place if you have changed rom version, ram size or loaded extensions in a different sequence. If all these things are unchanged you should be able to load images directly, controlling the process with a little machine code derived from MultiBasic. I leave that as an exercise for you, but I'm always interested to hear of suggestions for *DIY Toolkit* codes.

text87 version 300 offers today's state-of-the-art user-friendly environment for document production. With integrated spelling-checker and extensive support for highest quality printing possible on daisywheel and 9-pin printers.

fountext88 and **founted89**, the graphic printer-driver and editor for text87 provide graphic founts without the limitations in text editing and document size imposed by QL desktop-publishing programs. More than 35 quality founts of up to 72 pixels in height are supplied and new founts can be designed or captured from saved screen images.

2488, the state-of-the-art text-mode printer drivers for selected Epson LQ, NEC P, Star LC and Panasonic KXP 24-pin printers offer advanced features such as multiple typefaces, proportional spacing, micro-justification, double height, shadow and outline modes (depending on the printer model).

New Release
typeset90-deskjet New text-mode printer drivers for Hewlett-Packard Deskjet and Deskjet plus support printer's internal founts plus the full range of Roman, Helvetica and Gothic plug-in cartridges. **price: £20**

Software is available in English, French, German and Italian. Prices are inclusive of airmail. Payable by cheque, Postal Order or Eurocheque. Please specify language and disk system (all 31/2" and 51/4" formats can be supplied). text87 requires at least 256K memory expansion.

text87: £60 fountext88 plus founted89: £40 2488: £15
special offer extended to 15 July: £10 discount on orders of £100 or more

Software87, 33 Savernake Road, London NW3 2JU

Robin
Stevenson
continues his
Archive suite.

ARCHIVE POWER III

ROBIN STEVENSON ARCHIVE POWER Part Three Listing 1 Page 1

Listing One for Part Three

Listing One - Add these procedures to the main NOTEPAD program.

```

proc CALCSETUP                      : rem CREATE NEW CALCULATOR FILE
  create "CALCFILE" logical "CALC"
    QUESTION$
    ANSWER
  endcreate
  append
endproc

-----
proc CALCULATOR
  local C:CENPRINT;2,"CALCULATOR"
  error FOPEN;"File Is CALCFILE","CALC"      : rem OPEN CALC FILE
  if errnum()=100: error CALCSETUP: endif      : rem OR CREATE NEW ONE
  last : while 1                             : rem ***** MAIN INPUT LOOP
    paper 2:CENPRINT;0,"Answer : "+str(ANSWER,3,0): paper 0
    GETSTRING;"Enter question (or <B><N><M><ESC>)",QUESTION$
    if errnum()=27                           : rem IF ESC IS PRESSED
      CENPRINT;2,"<ESC>"                     : rem PRUNE DOWN FILE
      first : while count()>CALCMAX: delete : endwhile
      error SHUT;"CALC"                      : rem CLOSE & RETURN
      return : endif
    let C=instr("BNM",upper(ANSWER$))         : rem CHECK FOR OPTIONS
    if C=1: back : endif                      : rem STEP BACK DOWN FILE
    if C=2: next : endif                      : rem STEP TO NEXT RECORD
    if C=3                                    : rem STORE ANSWER TO M VARIABLE
      GETSTRING;"Store ANSWER to which MEMORY number?", "1"
      let ANSWER$="let M"+str(val(ANSWER$),2,0)+"=ANSWER"
      endif                                  : rem CONVERTS ANSWER$ FOR USE LATER
    if C=0 or C=3                             : rem FOR FORMULA OR M VARIABLE
      CENPRINT;1,"Calculating..."
      spoolon TEMPFILE$+"TEMP_PRG" export
      lprint "proc TEMP"                     : rem CREATE A TEMP PROCEDURE
      if instr(lower(ANSWER$),"let ")=1: lprint ANSWER$
        else : lprint "let ANSWER = "+ANSWER$: endif
      lprint "endproc"
      spooloff : error TEMPMERGE              : rem MERGE & RUN TEMP
      if errnum()                            : rem CHECK IT RAN CORRECTLY
        CENPRINT;2,"Not valid due to error "+str(errnum(),2,0)
        let ANSWER=0: else :CENPRINT;2,"CALCULATOR": endif
      if C=0: let QUESTION$=ANSWER$           : rem IF IT'S A FORMULA
        if not errnum(): append : endif : endif
      endif                                  : rem AND IF NO ERROR, ADD TO FILE
    endwhile                               : rem ***** END OF MAIN LOOP
  endwhile
endproc

-----
proc TODOLIST                      : rem HOOK FOR TODOLIST PROGRAM (NEXT MONTH)
  run object "TODOLIST"
endproc

-----
proc CALENDAR                      : rem HOOK FOR CALENDAR PROGRAM (SEE LISTING 2)
  run object "CALENDAR"
endproc

```

Over the last two months we have developed a notepad program using the Psion Archive programming language.

In the process a selection of toolkit procedures have been presented, which can be applied to a wide range of programming needs. This month we shall be using a number of these to provide two more Desktop Facilities to add to the suite – a calculator and a calendar. Because of memory constraints the calendar will be a separate program file, although Archive's ability to chain programs will hide this from normal use. The calculator, on the other hand, needs to be readily available from any part of the program. It is to be 'on-line' from any standard menu, so will need to be present on every program file.

The calculator facility comprises the bulk of **Listing One** and should be added to the existing NOTEPAD program. There are a number of design constraints to consider: it must not disturb the main work area of the screen, only using the menu portion. For memory efficiency, it needs to be compact, but it also needs to do rather more than the basic four maths functions. Power and flexibility are to be prized above simplicity of use. We will not impose a 'traditional' electronic calculator interface which, it must be said, would take a lot more programming. Instead we shall use the existing 'Psion mathematical standard', if such a thing can be said to exist.

Standard

This will be familiar to all *Archive*, *Abacus*, *Easel* and *SuperBASIC* users, and comprises the normal method of entering mathematical expressions in any of these programs. The spreadsheet input is the nearest to what we are aiming at, although without the grid presentation. Being *Archive*, the questions, and the answers will be stored on a sequential file, which can be browsed over, and edited for recalculation. On top of the standard '+ - * /' functions, full access to trigonometry, logarithms, powers, square roots, and brackets are to be made available, with floating point answers to the full 14 significant figures. In addition, an unlimited number of memory stores are to be possible, so that the answers to different calculations can be subsequently combined.

So how are we to pack such vast functionality into a small on-line procedure? The answer, fortunately, is that we don't! *Archive* has already provided it, and all we have to do is make it available. It is

an unfortunate shortcoming of Archive that a program cannot pass a user response on to the Archive interpreter, as though it was a program statement. However, we saw last month how easy it was to get a program to write, merge and run a new procedure, at run time.

This idea can be readily applied to turn a user's mathematical question into a new procedure. Thus if the user wishes to know what 2+2 is, our program could turn this request into a one line procedure, 'let ANSWER = 2+2. After running the procedure, the ANSWER variable would contain the result. Having got this far, it is exactly the same, as far as our program is concerned, if the user wants to know $\text{SQR}((56.89/(\text{PI}() \star 27.67)^2) + (\text{DEG}(\text{ATN}(73.457))))$. In both cases, the whole string is passed on to Archive, preceded by 'let ANSWER =', and wrapped up in proc/endproc statements. All we need to do is pass ANSWER back to the user.

Two problems spring to mind in particular, relating to user-friendliness. Firstly, how do you know what is possible? I abrogate my responsibility on this count,

on the grounds that if you know why you should want to make a calculation anything like the second one above, working out how, from the User Guide, should be a doddle. Suffice it to say that anything which is a valid Archive numeric expression will be valid input for the calculator.

The other problem is what is to happen if it's not a valid Archive statement. If we simply pass on the keyboard input, and present it in a program line, it could be any kind of rubbish. Fortunately, we can again rely on Archive. The TEMPMERGE procedure handles merging and running of the new procedure. If it fails to load, or fails to run, the error will simply be trapped back to our program. We can then pass on the error number to the user, who can use the list of errors in the *User Guide* to help identify what was wrong.

How then do we turn this kernel of an idea into a usable program? Firstly, the questions will be stored in a standard Archive file. The CALCSETUP procedure will create a suitable file, if one cannot be found. When you finish using the calculator – by pressing the ESCape key – the file

will be pruned down to keep it manageable. You will recall a global variable called CALCMAX in last month's initialisation program. The oldest entries are deleted, until COUNT() equals CALCMAX. Twenty entries should be enough to recall old entries without filling up micro-drives with redundant data. No entries are lost until you finish with the calculator, so all of a current session's entries can be recalled. The file itself is straightforward, with a string variable for the question, and a numeric one for the answer.

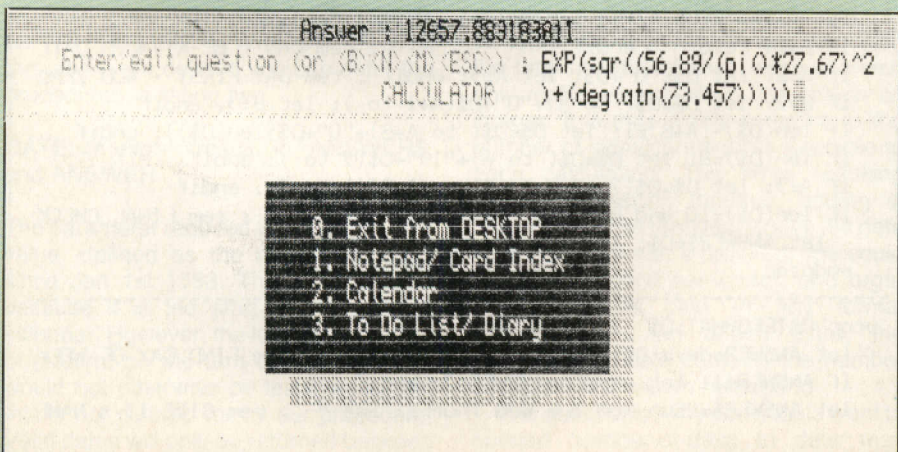
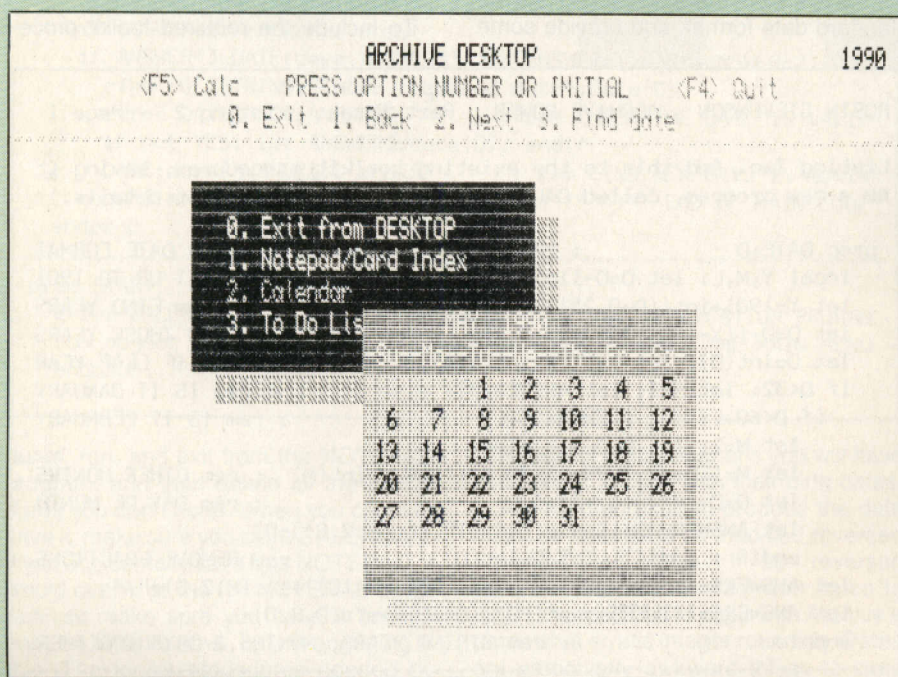
Interface

For the user interface, we shall solely use GETSTRING – developed two months ago. Provided a suitable SEDIT screen is available, this allows editing of existing formulae, as well as entering of new ones. As the calculator can be used from within other programs, such as NOTEPAD, it is important not to overwrite anything except the menu prompt area. The current question will be visible at the input line. The answer is to be displayed on the top line, and any error message on the line below.

One of the benefits of not using the more normal GETCHOICE procedure is the avoidance of any recursion problems. In providing facilities 'on-line' there is a great danger of escalating recursion – both wittingly and accidentally. Recursion is handy for the programmer, and it can save code duplication. But when it occurs, the code is still duplicated in memory, consuming large amounts of memory during use. The calculator – like the Quit function – is accessed through GETCHOICE, which contains direct and indirect recursion. After using a built in function direct recursion occurs when another copy of GETCHOICE is called. Indirect recursion could occur within the Quit function. It gets confirmation via CONFIRM, which asks its question via GETCHOICE, where you could again use the Quit facility. You can keep requesting Quit, running fresh layers of CONFIRM and GETCHOICE for as long as memory stocks last. Fortunately, this is an unlikely scenario, and any answer of no will cause the spiral to unwind, and most of the memory to be recovered. (Work out for yourself why answering yes causes no unwinding!)

The CALCULATOR avoids indirect recursion by not using GETCHOICE. If you wish to add other on-line functions you may need to address the problem, eg by setting a global variable, which makes the built-in functions unavailable to themselves. If your function needed to use the whole screen, you would also have to consider how to restore the screen to its original state, perhaps by keeping track of the screen in use, as we have with the file name.

One requirement, beyond simply asking individual questions, is the ability to store answers in memory, for use in



subsequent formulae. The method used is to select a memory number (for example, 3, 125, or 26). The current ANSWER is then assigned to that 'M' variable (for example M3, M125, or M26). Any M variable that has been assigned can then be used in a formula, as though it was a number. You can also use the ANSWER variable at any time, to incorporate the current answer.

For example, if M5 has been assigned a value, 'SQR(ANSWER+M5)' would be a valid formula. An M variable is assigned by entering a letter 'M' instead of a formula. You will then be prompted for the number you wish to assign, and this will be given the value currently in ANSWER.

Like formulae, the assignment requires a TEMP procedure to be created, so there will be a short pause before you can enter another formula. You can find out the current value of an M variable by entering it alone, as a formula (eg M5). ANSWER will then display its current value.

Another need is to browse around previously entered questions and answers. To do this, simply enter a 'B' for Back, or 'N' for Next, to move up and down the file. You can alter any entry, and pressing 'Return' will send the currently displayed entry to be calculated. Every entry that is calculated without error is added to the file, whereas invalid entries are returned for editing, without being stored.

A final feature is added for extra power and control. Normal entries are assumed to be just the formula, to which the program adds 'let ANSWER ='. However, if your question commences with 'let', the program assumes you have provided a total statement, and writes it on, unchanged. You can use this to assign a value to any variable directly, even variables being used by other parts of the DESKTOP, so use it with care.

Hooks

The final two procedures in the listing provide the 'hooks' for the Calendar and To-do list programs. There is a circular pattern to the way this works. When you select one from last month's DESKTOP procedure, the required program will be run. When you exit, the program re-runs the main NOTEPAD program again. This does not constitute recursion as Archive replaces the current program completely, making no attempt to store the existing program state. When you run a new program you cannot carry on where you left off afterwards. Instead, the new program must in turn run the old program, which will start from the beginning again. However, unlike SuperBASIC, the screen display, memory and file variables are unaffected, so it would be possible to communicate between the programs via global variables.

The Calendar procedures are printed in Listing Two. A fairly minimal provision is to be made, to which you could add your

own bells and whistles. We will show a month at a time, with the ability to move back or forward through the months, or find a particular one. The implementation is unusual for Archive, in that it uses no data files, requires to SEDIT screen, and produces no printed output. It is purely a handy on-screen reference facility.

Before we get to the nitty gritty of the calendar itself there are three new toolkit procedures to add. These are to fill in the gaps in Archive's date handling provision. Archive provides two date functions. It can return the date from the system clock, in three different string formats, and it can convert a given date into a numeric value, as the number of days since 1st Jan 1583. This 'DAYS' function, requires its parameter to be a date in the form 'YYYY/MM/DD', the same as that returned by the DATE(O) function. In our dealings with dates we shall regard this as the standard date format.

One function we need is to do the reverse of DAYS: that is, convert a numeric value (days since 1583) into a standard date string format. The second need is to ensure that a date string is in the standard date format, and provide some

conversion from other formats. Finally provision is to be given to display the date in a rather more readable form than that given by the standard date format. In addition, we will need to know the day of the week of a given date, and the number of days in a given month. These will be incorporated into two of the other procedures.

Toolkit

Having already produced a toolkit of procedures, we do not need to start from scratch for the calendar. Instead we can build on the foundations of the last two months. If you have memory expansion on your QL, you could simply add Listing Two to your existing NOTEPAD program, ignore all reference to program 'chaining', and replace the listing One CALENDAR 'hook' with the proper calendar procedure in listing two. However, an unexpanded QL attempting to do this would fail to run, due to insufficient memory. In this case you should provide the calendar as a separate program, which NOTEPAD will call automatically when selected.

To include the required toolkit proce-

ROBIN STEVENSON ARCHIVE POWER Part Three Listing 2 Page 1

Listing Two. Add this to the existing toolkit procedures, saving it as a new program, called CALENDAR. (See text for further details.)

```
proc DATE;D : rem CONVERT NUMBER-OF-DAYS TO DATE FORMAT
  local Y,M,L: let D=D-116146 : rem BRING IT UP TO 1901
  let Y=1901+int((D+0.25)/365.25-0.001) : rem FIND YEARS
  let D=D-((Y-1901)*365.25-0.75) : rem REMOVE DAYS IN THOSE YEARS
  let D=int(D): let L=(Y/4=int(Y/4)) : rem L=1 IF LEAP YEAR
  if D<32: let M=1: let ANSWER=31: else : rem IS IT JANUARY
    if D<60+L: let ANSWER=28+L : rem IS IT FEBRUARY
    let M=2: let D=D-31: else
      let M=((D+32.3999-L)/30.6): let M=int(M) : rem OTHER MONTHS
      let D=D-(M*30.6)+33.3-L : rem DAY OF MONTH
      let ANSWER=30+(instr("46911",str(M,2,0)))=0
    endif : endif : let D=int(D) : rem REMOVE FRACTIONS
  let ANSWER$=str(Y,2,0)+"/"+rept("0",(M<10))+str(M,2,0)+"/"
  let ANSWER$=ANSWER$+rept("0",(D<10))+str(D,2,0)
endproc : rem STRING YEARS, MONTHS & DAYS TOGETHER
```

```
proc DATECHECK;D$ : rem CONVERT DATE TO STANDARD FORMAT
  local A,B: let YES=0: let ANSWER$="" : rem FIND POSITIONS OF '/'s
  let A=instr(D$,"/"): let B=instr(D$(A+1 to ),"/")
  if B=0: return : endif : rem CAN'T BE A VALID DATE
  if A=2: let D$="0"+D$: let A=3: endif : rem DAY FIRST - ADD ZERO
  if B=2: let D$=D$( to A)+"0"+D$(A+1 to ): let B=3: endif
  if len(D$)-(A+B)=1: let D$=D$( to A+B)+"0"+D$(len(D$)): endif
  if len(D$)=8: let D$=D$( to 6)+"19"+D$(7 to ): endif
  if A=3: let D$=D$(7 to )+D$(3 to 6)+D$(1 to 2): endif
  if len(D$)=10 and B=3 and instr(D$,"/")=5 : rem FINAL CHECK
    let ANSWER$=D$: let YES=1: endif
endproc
```

```
proc DATEFORMAT;D$
  let ANSWER=days(D$)-(int(days(D$)/7)*7) : rem FIND DAY OF WEEK
  if ANSWER<1: let ANSWER=ANSWER+7: endif
  let ANSWER$="Sun Mon Tue Wed Thur Fri Sat " : rem GIVE IT A NAME
```



```

let ANSWER$=ANSWER$(ANSWER*4-3 to ANSWER*4)+" "+D$(9 to )+" "
let ANSWER$=ANSWER$+month(val(D$(6 to )))( to 3)+" "+D$( to 4)
endproc
: rem ADD DATE DETAILS TO STRING

```

```

proc CALDRAW;D$,MAX
local D,MON: let MON=val(D$(6 to 7)) : rem FIND MONTH
WINDOW;28,9,40,12: paper 2: ink 0: cls : rem SET WINDOW DISPLAY
print tab 7;upper(month(MON));" ":D$(1 to 4)
print "-Su--Mo--Tu--We--Th--Fr--Sa-"; : rem PRINT HEADINGS
print rept(chr(220),168);chr(18)+chr(1); : rem DRAW 'TILES'
DATEFORMAT;D$(1 to 8)+"01" : rem FIND STARTING DAY OF WEEK
paper 0: ink 7: print at 2,ANSWER*4-5;; let D=1 : rem GO TO IT
while D<=MAX: print gen(D,4);: let D=D+1: endwhile
WINDOW;80,25,0,0: print chr(18)+chr(0); : rem PRINT DAYS. RESET
let ANSWER$=D$ : rem RETURN THE DATE ACTUALLY USED
endproc

```

```

proc CALENDAR
local MAX:DATE;days(date(0)) : rem FIND INITIAL TOTAL DAYS
BOX;29,9,40,12,2 : rem RED BOX FOR CALENDAR
while 1 : rem ***** START OF MAIN LOOP
let MAX=ANSWER : rem MAX = NUMBER OF DAYS IN MONTH
error CALDRAW;ANSWER$(TO 8)+'14',MAX : rem DISPLAY THE MONTH
GETCHOICE;"0. Exit 1. Back 2. Next 3. Find date",4,"EBNF"
if ANSWER=0: return : endif
if ANSWER<3:DATE;days(ANSWER$)-30*(ANSWER=1)+30*(ANSWER=2)
else :GETSTRING;"Enter required date",date(0)
error DATECHECK;ANSWER$ : rem CHECK THE DATE IS VALID
if not YES: let ANSWER$=date(0): endif
DATE;days(ANSWER$): endif : rem FIND NEW DAYS IN MONTH
endwhile : rem ***** END OF MAIN LOOP
endproc

```

```

proc START
error CALENDAR : rem ERROR-TRAP THE APPLICATION PRORAM
run object 'NOTEPAD' : rem AND THEN RE-CALL THE MAIN MENU
endproc

```

dures, run, and exit from the NOTEPAD program, and then delete all the procedures you don't need. When you come to save it, make sure you call it CALENDAR. If you accidentally save it as NOTEPAD, it would overwrite the full NOTEPAD program, so make sure you have a backup copy on another drive, before engaging in this. The toolkit procedures needed by CALENDAR are:- BOX, CALCSETUP, CALCULATOR, CENPRINT, CONFIRM, FOPEN, GETCHOICE, GETSEEDIT, GETSTRING, SHUT, WINDOW. Once all the others are deleted you can enter the procedures in listing two.

DATE;<n.exp> :result in ANSWER\$, and ANSWER

The parameter required is a numeric date value, defined as the number of days since Jan 1st 1583. This date is used because it is the start of the modern calendar. However, the leap years fall out of pattern, on the turn of centuries that would not otherwise be leap years, and this is not catered for by our procedure. Valid dates will only be returned between 1901 and 2100, so if you are compiling a

database of nonagenarians you will have to modify it to cope with their birth dates.

The method used to produce the date might be described as modified reversed Schecter technique. It is reversed because the original was from dates to days. It is modified to cope with Archive's somewhat erratic integer rounding. Dates are particularly awkward things to cope with mathematically. This method depends on using a number of empirically derived constants, which can be used to calculate the year, month, and date from the total number of days, each of which only makes sense when reduced to an integer. It is thus highly sensitive to the number of significant figures available to each calculation. The three resulting values are strung together, adding in leading zeros as required, to give the date in standard format. If required, you could include a second parameter, and organise the final date in any format requested. As an added extra, the ANSWER variable contains the number of days in the month referred to.

The question of why you should want to convert number-of-days to date may seem as obscure as how it achieves it.

The origin of the number of days will almost certainly be the Archive DAYS function, since it is not something you normally know. Its power is in the ability to perform arithmetic using dates. If you have a date, D\$, and need to know the date for five days beyond that, you could use DATE;days(D\$)+5. All the month-ends, leap years, year-ends and so on are handled correctly, and the required date is available in ANSWER\$.

DATECHECK;<s.exp> : result in ANSWER\$ and YES

If we are to have a standard date format, for use with other procedures, it is essential to check newly entered dates, to ensure they conform to the format. It is also highly desirable to allow users some flexibility in the format they can enter. The DATECHECK procedure helps satisfy both these needs. A date string is passed as the parameter, which DATECHECK attempts to handle correctly. If it can be turned into the standard format, ANSWER\$ will contain the result and YES will be true. If DATECHECK finds an error in the format, ANSWER\$ will be empty, and YES will be false. So what are the allowable input formats?

Formats

There are two formats supported, corresponding broadly to the Archive DATE(0) and DATE(1) formats. However, DATECHECK does not need the leading zeros for month or day - it will insert them automatically. In addition, if using the day first format of DATE(1), it will default to the 20th century if only two year numbers are supplied. If you are referring to next century, or wish to enter the date with year first, all four year numbers must be included. Thus the dates 25/6/90, 25/06/90, and 1990/6/25 are all acceptable, and will all result in ANSWER\$ equalling '1990/06/25'. Error checking, while extensive, is not total and it is possible to enter non-numeric values which are not caught.

If this is likely to be a problem, you could add further checks. Nor does it catch numbers that don't comprise a sensible date. '1990/05/56' would be acceptable, not just to DATECHECK but also to Archive's DAYS function! The main use will be after obtaining a date entry. Afterwards ANSWER\$ should be assigned back to the required variable. It is up to the calling procedure whether a wrong entry is to be left blank, substituted for an alternative, or re-entered.

DATEFORMAT; <s.exp> : result in ANSWER\$ and ANSWER.

The purpose of the DATEFORMAT procedure is to make dates more readable and informative than as provided by the standard date format. If it looks familiar, that's because we used similar code in the

DESKTOP initialisation routine. Whereas that took the system clock as its input, here you can pass any date in the standard date format, as the parameter. The format chosen returns an abbreviated day of the week, the day, an abbreviated month, and the year, ending up with a 16 character string in ANSWER\$. In addition DATEFORMAT needs to calculate day of the week, and so this is also returned in ANSWER\$, as a number between 1 and 7, Sunday being day 1.

With the dates toolkit now complete, we can readily produce a quick and easy calendar program. For convenience it is broken down into two procedures, one to draw a given month on the screen, the other to handle interaction with the user.

CALDRAW;>s.exp<,>n.exp<

The drawing routine requires two parameters. The first is a date string, in the standard date format, although only the year and month are used. The other is the total number of days in the month. CALDRAW could use DATE to find this information, but CALENDAR has already done so, and it saves time not to calculate it twice. Since the sole purpose of the calendar is a screen image, we might as well make it a good one.

We shall therefore employ various tricks to create an interesting 'page' from

a calendar. A month can span a maximum of six weeks, so we shall need six lines for the dates. To allow four spaces per day, a block 28 characters wide will be needed. The effect we are aiming at is a tiled block, with the numbers right-justified on their tiles. A window 28 characters wide is used, so that, after drawing the month and day headings, the tiles can be quickly drawn. Printing 168 stipple/tile characters fills the 6 week block, and changing the print attribute to transparent paper (using the printer driver character 18) allows the numbers to print on top of the tiles. DATEFORMAT is used to find the day of the week for the first, and the cursor is positioned accordingly.

From then on, each day can be printed, formatted in a four character field with GEN(D,4) without further need to position the cursor. Once printing is complete, the full window can be restored, print attributes set to normal, and the value in ANSWER\$ restored to the date originally set.


Control

Finally we can add the control procedure for the CALENDAR. Because so many of the procedures it deals with return the current date in ANSWER\$, we shall use that variable directly, and save a lot of re-assigning. You must be sure that none of the procedures overwrite your

global variable, when doing this. A rather obscure programming method is used to control movement from one month to the next. When selected, 30 days are either added to, or subtracted from the current date, using DATE. Unchecked, this could cause some months to be missed out completely. To remedy this, the date is restored to the 14th of the month, every time the calendar is redrawn.

Once you are happy the calendar is functioning correctly, and is properly saved, you should add the START procedure, which links it into last months NOTEPAD program. Having saved it again, you can then run DESKTOP and see how transparently you can move from NOTEPAD to CALENDAR and back again.

It would be possible to expand from this calendar idea. If you were to incorporate a data file, you could mark certain days or periods - for example as a holiday planner for a group of employees; a project planner, plotting the various stages of a project; or incorporating diary elements, allowing access to individual weeks or days as well as months. If that is what you need, you should now have the means to develop it. Next month we stay with dates, but take a rather different approach. We shall be developing a to-do list/appointments diary, based not on monthly pages, but on scrollable lists, with any number of entries.



EEC LTD

MAIN SUPPLIERS OF SINCLAIR QL COMPUTERS & LOW-COST PRINTERS - QLs From £90 PRINTERS From £80

BACKUP QL the QL with Power Supply Unit & TV Lead. Keep it as insurance, or for networking, cheaper than changing to another system. **JM ROM £90**

QLs COMPLETE Fully tested and with 3 months warranty. TV lead, QL Software 2.35. Also complete bound user guide for the QL, SuperBasic & for the software: Quill (word processor) - Abacus (spreadsheet) - Archive (records) - Easel (business graphics). **JS ROM £100**

CUSTOMERS BUYING ONE OF THE COMPLETE UNITS ARE GIVEN ONE YEAR'S FREE MEMBERSHIP TO QUANTA. (Help, Newsletters & 400+ Library Programs - most Free). **JM ROM £120**

QUANTA MEMBERS CAN OBTAIN A £5 DISCOUNT **JS ROM £135**

NEW ITEMS

1MB 3.5" DISC DRIVE UNIT with front bezel.....£49

TANDATA MODEMS, Q-MOD, Q-CONNECT, Q-CALL, £20 each, or the Set.....£40

ACCESSORIES & SPARES

| | |
|----------------------------------------------------------------------|----------------------------------|
| QL Psion software 2.35 pack £20.00, or separate programs each £10.00 | |
| Complete bound user guide.....£7.00 | QL power supply unit.....£12.00 |
| Joysticks.....£9.00 | JM ROMs sets.....£10.00 |
| TV or Network leads.....£3.00 | Centronics interface.....£24.00 |
| MC 68008 microprocessor.....£8.00 | Serial printer lead.....£12.00 |
| Keyboard membrane.....£9.50 | ZX 8301 or ZX8302 each.....£9.00 |
| Cartridges, wallets of 4, new £12.00 or unused prog cartridges £9.00 | |
| 20 new cartridges in plastic storage box..... | £55.00 |

SOFTWARE

| | |
|---------------------------------------------------------|------------------------------------------------------------------------|
| Cash Trader, Decision Maker, each.....£18.00 | QL Liberator SuperBasic Compiler.....£60.00 |
| Touch & Go (Typing Tutor), QL Games Cartridge.....£9.00 | QLoad SuperBasic: Quickloader with QRef programming utility.....£15.00 |



BUSINESS SOFTWARE

| | |
|--------------------------------------------------|-------------------------------------------------------|
| PDQL Payroll.....£80.00 | PDQL Mailmerge de Luxe.....£20.00 |
| PDQL Trading Accounts (inc Analyser).....£125.00 | Send stamped addressed envelope for Software leaflets |

LOW-COST PRINTERS

With instructions for Quill, Abacus & Archive

| | |
|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--|
| SEIKOSHA GP100A Centronics, 80 col, 50 cps, graphics, tractor feed, Centronics interface.....£20.00 | |
| (if bought with printer) | |
| C. ITOH serial daisywheel printer, prints at 20 cps, can print 1+3 copies. Paper width 13ins. Chars per line: 118, 135 or 172 (selectable on top panel). Buffer 256 bytes. Obtainable extras: Centronics module, 2-col ribbon, tractor and auto sheet feeder. Alternative plug-in interface modules for Star NL-10 printer. Serial, parallel/Centronics, IBM/PC or Commodore, each.....£20.00 | |






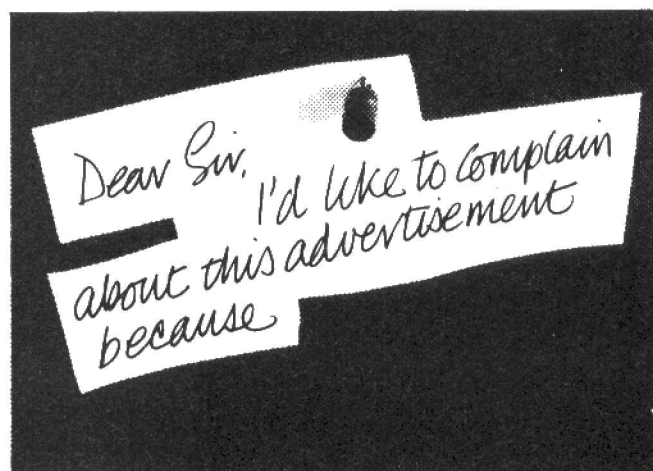
TERMS CWO, ACCESS or VISA

Minimum order £9. P&P £6 for printers & QLs. (Overseas £15). Other items £3. (Overseas £8)

EEC LTD

18-21 MISBOURNE HOUSE, CHILTERN HILL,
CHALFONT ST PETER, BUCKS, SL9 9UE.
TEL: 0753 888866 FAX 0753 887149



Most advertisements are legal, decent, honest and truthful. A few are not, and, like you, we want them stopped.

If you would like to know more about how to make complaints, please send for our booklet: 'The Do's and Don'ts of Complaining'. It's free.

The Advertising Standards Authority.
We're here to put it right.

ASA Ltd., Dept. Z, Brook House, Torrington Place, London WC1E 7HN.

This space is donated in the interests of high standards of advertising

SOFTWARE FILE

INFORMATION:

Product: QPAC 2

Description: QDOS management system

Price: £49.91 (£29.90 as upgrade from QRAM)

Distributor: Care Electronics, 800 St Albans Road, Garston, Watford, WD2 6NL.

QPAC-2

Mike Lloyd looks at a much-improved utility upgrade.

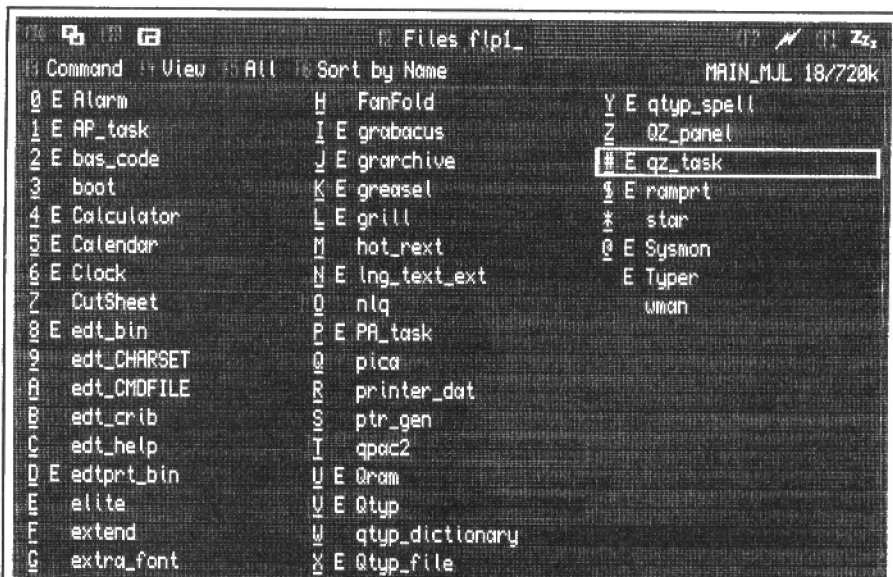
Although from the very beginning the Sinclair QL was described as a multi-tasking computer, the only hint of this impressive capability at its launch was a tiny listing for a multi-tasking digital clock routine in the *QL User Guide*. Typically for the QL at that time, it failed to work. Nevertheless, third party suppliers soon latched onto the real potential of the QDOS operating system even though it was incomplete when the QL went on sale and was only partially improved by the time that Amstrad effectively killed off further development.

Tony Tebby, through QJump, has been the leading exponent of making the most of QDOS since Amstrad bought Sinclair. This is hardly surprising as he was the original author of QDOS. The developments have followed two distinct paths. Firstly Tebby improved QDOS's functionality through *Super Toolkit* and *Super Toolkit 2*, and more spectacularly he tamed the excesses of untrammelled multi-tasking with his QRAM suite.

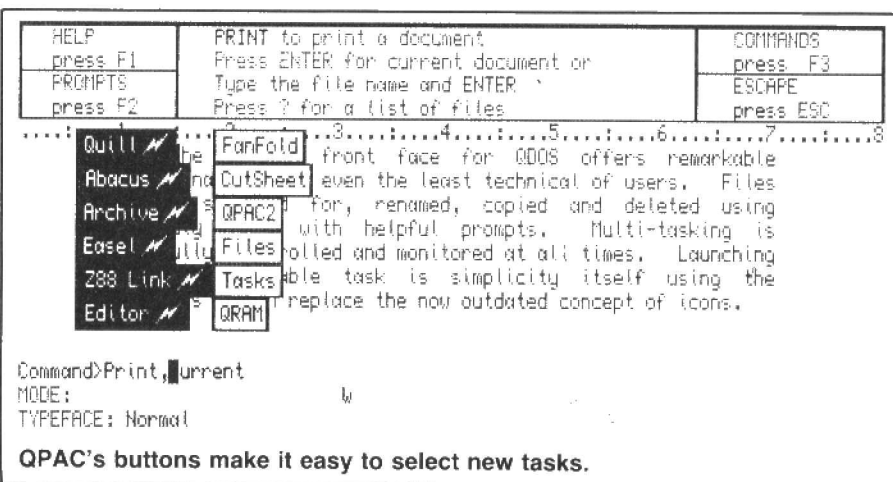
By contrast, QJump's *QPAC 1* package, a miscellany of desktop utilities such as calculator and calendar, has not been such a success. Recently, however, with hardly any publicity whatsoever, Tony Tebby has released a replacement for QRAM under the name *QPAC 2*. For the modest cost of this substantial piece of software, QL owners are going to think that they have bought a new computer.

QPAC 2 is delivered in a smart ring blinder with a 3.5in. disk tucked into the front pocket and containing a manual of some 80 A5 pages. For a first look at the system it is best to load the disk and let the default boot program do its job. In response to customer demand Tebby has developed a tutorial program designed to introduce newcomers to the wonders of QPAC 2 without over-awing them. Unfortunately, the version on the very early copy of QPAC which I purchased had a minor glitch and did not work properly, leaving me more confused than I was before I started. The problem should be well and truly sorted out by now.

Like QRAM, QPAC 2 manages files, devices and tasks. To do this it sets up what Tony Tebby calls the "Extended Environment", a mixture of a pointer



A typical file menu with 4 programs selected.



QPAC's buttons make it easy to select new tasks.

interface, a window manager and a hotkey manager. Early pointer interfaces were either keyboard- or mouse-specific, but this is no longer the case and the "Ptr-gen" general utility operates happily under the control of either device. Mice being rather rare in the QL world, it is pleasing to note that the improved interface ensures that the pointer moves quickly, smoothly and accurately under the control of the cursor keys. Pressing the space bar or the Enter key produces a response similar to clicking

the left or right mouse button.

The window manager is the most revolutionary part of the extended environment, although it will be familiar to QRAM users. Imagine that instead of your QL having only one monitor screen it has several stacked one behind the other, rather like a pack of cards. With a keypress the screen at the bottom of the pile can be brought to the top and displayed while the program to which it is

cont.

QDOS EXTENSIONS PROVIDED BY QPAC 2

The following commands create "buttons" linked to tasks. The task name, button text, location co-ordinates and colours can all be specified by parameters.

BT_SLEEP BT_EXEC BT_WAKE BT_HOTKEY

The following functions link tasks with 'hotkeys' (ALT plus an alphanumeric).

HOT_WAKE HOT_THING HOT_KEY HOT_CMD
HOT_RES HOT_RES1 HOT_CHP HOT_CHP1
HOT_LOAD HOT_LOAD1 HOT_PICK

The following procedure allows any of the above functions to be used as stand-alone commands rather than in expressions.

ERT

The following procedures execute tasks with optional parameters for Psion programs, impure code, unlocked windows and guardian windows.

HOT_LOAD EXEP

Other miscellaneous keywords.

HOT_LIST HOT_SET HOT_OFF HOT_REMV
HOT_DO HOT_STUFF HOT_NAMES HOT_TYPE

attached ceases to be dormant and becomes the "current" task. Any executable job can join the stack, even multiple copies of interpreted SuperBasic programs for owners of the more recent releases of the Minerva Eprom. Where the program windows are smaller than the full screen many tasks can be displayed simultaneously, although normally only one is active at any one time.

This is the key difference between QPAC 2 and normal QDOS: QPAC 2 is by default a *concurrent* environment in which many programs exist simultaneously but only one is working at any one time, whereas QDOS is a *multi-tasking* environment in which many tasks can be active at one time, although only one of them can accept keyboard input. This distinction is slightly blurred by QPAC's ability to "unlock" windows so that they continue to be updated even though another task's window is accepting input.

QPAC 2 can be loaded and begin working without anything appearing on the screen to betray its existence. However, by pressing a pre-defined hotkey (in conjunction with the ALT key) a number of "buttons" appear on the screen along with an arrow pointer. Buttons are tiny windows just large enough for some text, perhaps a couple of short words, which describe the program which they represent. By means of the cursor keys the pointer can be manoeuvred to any of the buttons and, by pressing the Enter key, the program which is linked to the button is awakened and becomes the current task.

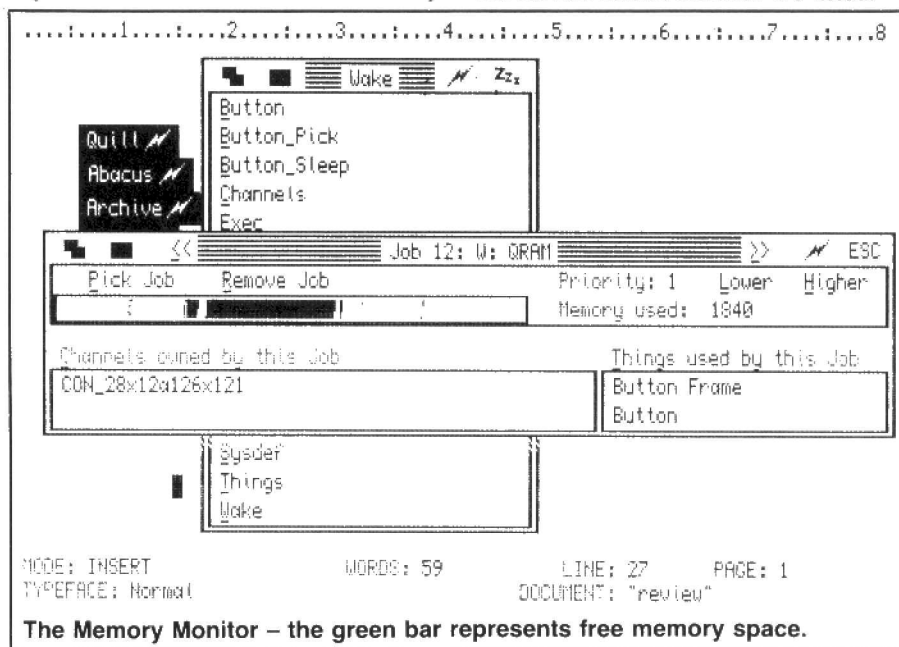
Further buttons can be pressed to make other selections, until the QL's memory is full. Tasks can of course be removed, thus releasing memory for further jobs to take their place. What makes this system all the more impressive is that buttons can be used to wake any executable programs, be they QPAC utilities, the old QRAM

menu, the Psion packages or SuperBasic programs compiled with *Turbo* or *QLiberator*. If moving pointers across screenfuls of buttons is too much like hard work the Extended Environment allows you to select the same tasks by pressing a defined hotkey or by picking a job from a menu.

QPAC buttons come in four flavours, allowing users to design their own system management environment according to their particular needs. BT_SLEEP is used to declare a button linked to a task which is loaded into the QL but not activated until the button is pressed. BT_WAKE works in a similar way but the button remains available even when the task it belongs to has been activated, allowing more than one copy of its task to be called into existence at one time. BT_EXEC is used to call up a task from the default drive, making it an ideal way of loading a Psion program. Finally, BT_HOTKEY provides a way of using the pointer interface rather than an ALT-key combination to activate a hotkey

of the icons found on most graphical user interfaces (GUIs). As was recently pointed out, it might be clever to draw a waste-paper basket on the screen to represent the delete command but to a whole generation of Americans wire baskets are not items of office furniture with which they are familiar. The conceit of icon-ridden GUIs is that they assume users cannot read. QPAC 2 is not so condescending. If you want to call up a program called Quill, hit the button marked "Quill", not a stylised picture of a piece of paper.

The major part of QPAC 2 both in terms of code space allocated and in its usefulness to QL owners is the file management window. From here all of the tedious but essential housekeeping jobs relating to files, disks and microdrives can be undertaken using a few keystrokes. The window is divided into the heading area which displays information and itemises some of the available commands, and the main display area in which all files on the current source medium are listed.



The Memory Monitor – the green bar represents free memory space.

sequence (which might itself execute a task or do something more prosaic, such as write an address into the "current" word processor).

To make it easy to find all these buttons, QPAC 2 includes a task called the Button Frame which responds to a user-defined hotkey. All buttons can be loaded into the frame so that they can be displayed simultaneously and in good order. Because the button frame has a similar function to the old QRAM main menu I used the familiar ALT- combination as the hotkey. The button frame itself can be configured to use any part of the screen and the user can opt for the buttons to fill the frame horizontally or vertically. Users should be warned that placing buttons in the button frame can vary their functionality and that there is still a role for "free-floating" buttons.

Buttons are the hype-free equivalents

The list of files can be sorted in ascending or descending order using file-name, size, date created or time created, singly or in combination, as criteria. If there are more files to list than there is room for, the window shows vertical or horizontal bars to indicate where in the overall scheme of things the window is currently situated. The bars can be dragged with the pointer in a manner which will be familiar to users of other graphical

QPAC 2 UTILITY TASKS

BUTTON_PICK
Displays all buttons in the 'button frame'.

BUTTON_SLEEP
Puts a task "to sleep". Its windows are taken from the stack and replaced by a button. Press the button to awaken task.

CONFIG
Sets defaults for QPAC 2 and for any other program using the same configuration standard

OPAC 2: MAJOR MENUS

SYSDEF

Sets defaults such as directories, print destination and pointer speed for the current session.

THINGS MENU

A thing, apparently, is a task with certain characteristics. What they are Tebby is not saying. From the Things menu you can reach a status window showing the QDOS extensions and jobs owned by a selected Thing.

EXEC MENU

Lists tasks which can be executed. A parameter string can be declared if required.

WAKE MENU

Lists sleeping tasks which can be woken.

JOBS MENU

Lists all current jobs. A submenu shows the things and channels associated with the job and a graphic display of RAM usage.

PICK MENU

Lists current jobs owning windows. Jobs can be "picked" to the top of the window stack and activated.

RJOB MENU

Allows any job except SuperBasic and RJOB to be removed from memory.

HOTKEYS MENU

Lists and activates user-defined hotkeys.

HOTJOBS MENU

Lists only those hotkeys linked to tasks.

CHANNELS MENU

Lists and permits closing of channels

interfaces and which should be intuitive to QL users meeting the system for the first time.

window pops up under the cursor giving the choice of moving, copying, viewing, executing, updating, printing and deleting the file.

File activity is closely associated with the Super Toolkit equivalents. For example, copying a group of selected files from the source directory to the destination directory produces exactly the same prompts as those used in Super Toolkit, albeit ensconced in their own pop-down windows. One gets the feeling that this sort of friendly dialogue between software and computer using is not entirely to Tebby's liking. "A front end", he writes, "provides a distorted view of the computer's operation." Whatever the views of the author, the software meets the needs of all types of users, experienced or novice, in an efficient, flexible and trustworthy manner.

The options on the file menu's header are each preceded by a function key name by which they can be selected. It is usually easier, however, to use the pointer to highlight the required option. The file menu's window can be moved around the screen, resized or closed. The source directory can be altered to list files on any medium or any subdirectory on that medium. The F3 key calls up the file commands window, consistent with the Psion menus. Other function keys access the sorting options, allow file contents to be viewed and allow all files to be selected

The directories menu makes a stab at making sense of the QL's weak understanding of subdirectories. Presumably with the maximum 100K limit of a microdrive in mind, QDOS lacks the hierarchi-

The best advice I can give to anyone who has attempted to establish a consistent set of directories on a disk is to be patient and to experiment with OPAC. Hard disk owners are in a much stronger position because, thankfully, it was recognised that the Version 1 device drivers were inadequate when faced with 40Mb of storage space. OPAC 2 is completely at home managing hard disk activities.

OPAC 2 comes with some additional utility programs on the medium, the most useful of which is the "ramprt" file which was originally bundled with the QRAM program. This binary file sets up the fastest RAM disks around (a RAM disk is a notional device which uses part of the QL's random access memory to emulate a microdrive or disk) and also establishes a print spooler (a device which forces print jobs to queue up until the printer is free). Much to my regret, there was no room for the excellent window dump utility which was such a valuable part of QRAM. QRAM owners should therefore install QRAM as part of OPAC 2. They will find that the new window manager adds extra functions to all of the old QRAM windows.

As one of the first people to obtain OPAC I have been using it for some months now and quite honestly the QL would only be half the beast without it. There are a very few programs which should be part of everybody's collection. *Lightning*, of course, is an essential, as is Super Toolkit 2. Add to these two outstanding utilities OPAC 2 and your computer can confidently look even the latest breed of PCs in the eye as an equal.

THE OPAC 2 FILES MENU

Displays all files matching a wildcard such as "ram1.doc" or simply "mdv1".

Files can be sorted according to name, size, date of creation or time of creation.

Multiple sorting criteria (eg date & name) can be specified.

All listed files, or selections from the list, can be selected to be operated on by the following commands:

VIEW. The contents of each file are displayed, in ASCII, in a scrolling window

COPY. Each file is copied to the destination directory. Conflicts are resolved with the *Overwrite?* (Yes, No, All, Quit) prompt.

MOVE. Each file is copied as above, the original file being removed from the source.

UPDATE. Each file is copied as above provided that (a) a file of the same name exists on the destination directory, and (b) the source file is dated more recently than its match in the destination directory.

BACKUP. Each file is copied provided that a more recent file of the same name is not on the destination directory.

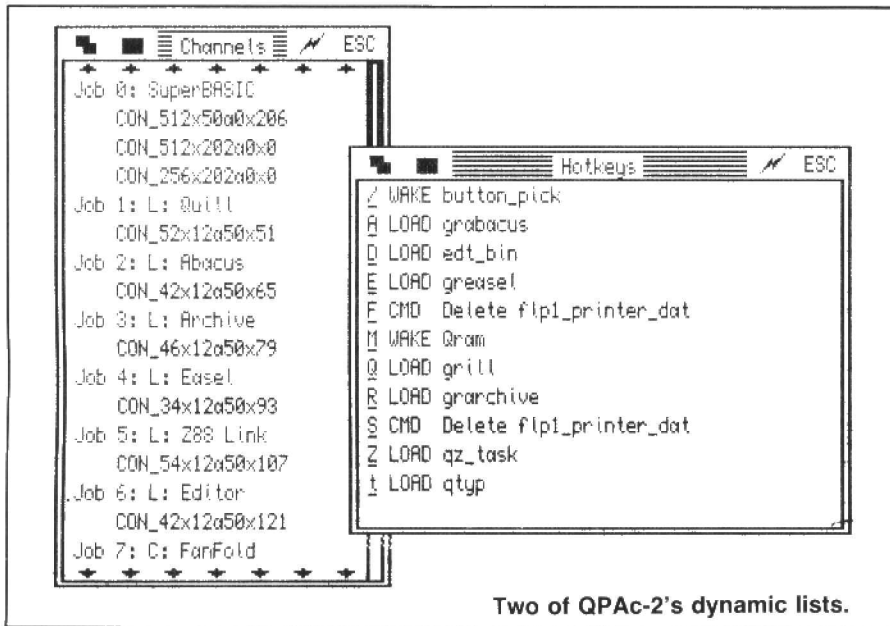
DELETE. Each file is deleted from the source.

PRINT. Each file is printed provided that it is in a suitable format.

FORMAT. The destination medium is formatted.

STATISTICS. Adds the size, type and time/date of creation to the file list display.

TREE. Inhibits directory wildcard.



Two of QPac-2's dynamic lists.

As the pointer moves across the list of files each in turn is highlighted by a white box. Pressing the space bar has the effect of "selecting" the file so that by selecting a number of files you can carry out an operation (deletion, for instance) on them all with a single command. If the Enter key is pressed instead of the spacebar a

cal file storage structures familiar to users of MS-DOS and Unix. The Psion programs certainly made the unfortunate assumption that a medium shall comprise but a single directory. Super Toolkit 2 allows subdirectories to be declared in file operations and OPAC 2 attempts to find them.

This program is offered as a study for those who like word puzzles. The idea is to get someone to type in a message of reasonable length, while you go and make a cup of tea. The program encodes it and you must then decode it without the key.

The program enables you to do this on the screen (no pencil and paper required) and it displays a letter frequency table to save you the bother of counting them up. The program runs on a monitor or on a TV Screen, so if you 'cheat' when the program asks you about this, you will get a somewhat garbled display.

A word or two about deciphering codes: all of the ciphers produced by the program are simple substitution codes. That is, every letter of the initial message is nearly always substituted for another in the alphabet. Numbers, punctuation marks and foreign letters are not encoded. Every run of the program produces a different code.

To decipher a coded message you must be aware that certain letters appear more frequently than others. In the

P + R : O = G < S

If you have a program worthy of consideration, send it to 'The Progs', Sinclair QL World, Panini House, 116-120 Goswell Road, London EC1V 7QD. We pay for everything published at the usual rates.

CIPHERS

by C. B. Storey

English language E is, by far, the most common, followed by T A I S O N, in that order. Of course, this is not necessarily true in any particular text. And some authorities favour E T A O N I R S H.

So, if there are more Xs than anything else in the encoded message, X is likely to be E. Your vocabulary and your knowledge of grammar and syntax will help you to make intelligent guesses about the

rest of the coded message and, in the end, solve the puzzle.

But if you get really stuck, you can enter '99' to see the original message in clear.

It makes a pleasant change from crosswords.

```

100 REMark *** 'CIPHERS' (c) C B Storey 1990 ***
110 CLEAR:PAPER 0:INK 7:CLS:CLS#0
120 POKE_W 163976,256:REMark U/CASE
130 mon_tv
140 REPEAT main
150 CLS:CLS#0:centre 1,10,"One moment...":b2
160 a$="ABCDEFGHIJKLMNOPQRSTUVWXYZ"
170 DIM c$(26),freq$(26,6)
180 FOR n=1 TO 26:freq$(n,1 TO 6)=a$(n) & "= 0"
190 FOR n=1 TO 26:c$(n)=" "
200 compile_code:CLS
210 INK 2:centre 1,0,"C I P H E R S"
220 INK 7:centre 1,2,"Type in your message for encoding"
230 centre 1,4,"Press ENTER at end of message"
240 INK 2:centre 1,6,"(No more than 5 lines of text allow
250 INK 4:AT 8,0
260 get_message
270 STRIP 7:INK 0
280 centre 1,19," Encoding message...please wait "
290 STRIP 0:INK 7
300 tri$=m$
310 FOR n=1 TO LEN(tri$)
320 IF tri$(n) INSTR a$
330 tri$(n)=" "
340 END IF
350 END FOR n
360 check_string:encode

```



```

370 CLS:INK 2:AT 1,0:PRINT enc$
380 STRIP 7:INK 0
390 b2
400 centre 1,19," Preparing frequency table...please wait "
410 STRIP 0
420 freq:instns:dialog
430 CLS#0
440 INK 7:AT 0,0:PRINT m$:b2
450 STRIP#0,4:INK#0,0
460 centre 0,1," Press 'R' to Run again; or 'Q' to
470 REPEAT rq
480 rq$=INKEY$(-1)
490 IF rq$ INSTR "RQ":EXIT rq
500 b5
510 END REPEAT rq
520 STRIP#0,0:INK#0,7
530 IF rq$="Q"
540   CLS:CLS#0
550   POKE_W 163976,0:REMark 1/case
560   centre 1,10,"Program STOPped"
570   EXIT main
580 END IF
590 END REPEAT main
600 b2:STOP
610 REMark *****
620 DEFINE PROCEDURE mon_tv
630   PAPER 0:CLS:INK 7:b2
640   REPEAT mtv_loop
650     AT 0,0:PRINT "Are you using a Monitor or a TV? ":PRINT
660     INPUT "ENTER 'M' or 'T' ":mtv$
670     IF mtv$="" :b5:CLS 3:NEXT mtv_loop
680     IF mtv$(1) INSTR "MT":EXIT mtv_loop
690     b5:CLS 3
700   END REPEAT mtv_loop
710   IF mtv$="M":wid=85:mon_window:ELSE wid=74:tv_window
720   END DEFINE mon_tv
730   REMark *****
740   DEFINE PROCEDURE b2
750     BEEP 2000,20
760   END DEFINE b2
770   REMark *****
780   DEFINE PROCEDURE b5
790     BEEP 5000,50
800   END DEFINE b5
810   REMark *****
820   DEFINE PROCEDURE compile_code
830     RANDOMISE
840     FOR n=1 TO 26
850       r=RND(1 TO 26)
860       IF c$(r) <> " ":n=n-1:NEXT n
870       c$(r)=a$(n)
880     END FOR n
890   END DEFINE compile_code
900   REMark *****
910   DEFINE PROCEDURE centre (chan,l,p$)
920     AT#chan,l,(wid-LEN(p$))/2:PRINT#chan,p$
930   END DEFINE centre
940   REMark *****
950   DEFINE PROCEDURE get_message

```




```

960 count=0:m$="":w$="":adj=0
970 REPEAT loop
980 IF count >= wid * 5
990   b5
1000  STRIP 2:INK 0
1010  centre 1,19," Message truncated "
1020  STRIP 0:INK 4
1030  PAUSE 80
1040  EXIT loop
1050 END IF
1060 z$=INKEY$(-1):BEEP 50,50
1070 IF CODE(z$)=10:b2:EXIT loop
1080 IF z$=" "
1090   spos=count+1
1100   w$=""
1110 ELSE
1120   w$=w$ & z$
1130 END IF
1140 m$=m$ & z$
1150 IF count MOD wid=0
1160   IF count > 1
1170     IF z$=" "
1180       m$=m$(1 TO LEN(m$)-1) & CHR$(10) & CHR$(10)
1190       PRINT:PRINT
1200       z$=""
1210     ELSE
1220       m$(spos+1+adj TO)=FILL$(" ",LEN(w$)-1) & CHR$(10)
1230       m$=m$ & CHR$(10) & w$
1240       AT 8,0:PRINT m$;
1250       adj=adj+2
1260       count=count+LEN(w$)
1270     'NEXT loop
1280   END IF
1290 END IF
1300 END IF
1310 PRINT z$;
1320 count=count+1
1330 END REPEAT loop
1340 END DEFINE get_message
1350 REMARK *****
1360 DEFINE PROCEDURE check_string
1370 FOR n=1 TO LEN(m$) STEP wid+2
1380 IF m$(n)=" ":m$=m$(1 TO n-1) & m$(n+1 TO)
1390 END FOR n
1400 END DEFINE check_string
1410 REMARK *****
1420 DEFINE PROCEDURE encode
1430 enc$=""
1440 FOR n=1 TO LEN(m$)
1450 IF m$(n) >= "A" AND m$(n) <= "Z"
1460   enc$=enc$ & c$(search)
1470 ELSE
1480   enc$=enc$ & m$(n)
1490 END IF
1500 END FOR n
1510 END DEFINE encode
1520 REMARK *****
1530 DEFINE FUNCTION search
1540 FOR x=1 TO 26

```




```

1550 IF m$(n)=a$(x):RETURN x
1560 END FOR x
1570 END DEFine search
1580 REMark *****
1590 DEFine PROCedure freq
1600 REMark *** Prepare freq$ ***
1610 FOR x=1 TO LEN(enc$)
1620 IF enc$(x) >= "A" AND enc$(x) <= "Z"
1630 freq$(CODE(enc$(x))-64,4 TO 6)=freq$(CODE(enc$(x))-64,4 TO 6)+1
1640 END IF
1650 END FOR x
1660 REMark *** Sort freq$ by frequency ***
1670 FOR x=1 TO 25
1680 flag=0
1690 FOR y=1 TO 25
1700 IF freq$(y,4 TO 6) < freq$(y+1,4 TO 6)
1710 temp$=freq$(y,1 TO 6)
1720 freq$(y,1 TO 6)=freq$(y+1,1 TO 6)
1730 freq$(y+1,1 TO 6)=temp$
1740 flag=1
1750 END IF
1760 END FOR y
1770 IF NOT flag:EXIT x
1780 END FOR x
1790 REMark *** PRINT sorted freq$ ***
1800 CLS 3
1810 INK 2:centre 1,12,"FREQUENCY TABLE (Memo: ETAISON)"
1820 l=13:c=6
1830 INK 7
1840 FOR x=1 TO 26
1850 AT l,c:PRINT freq$(x,1 TO 6)
1860 l=l+1
1870 IF l=16:l=13:c=c+8
1880 END FOR x
1890 END DEFine freq
1900 REMark *****
1910 DEFine PROCedure instns
1920 INK 2:centre 1,16,"SOLVING DIALOGUE (by example)"
1930 INK 7
1940 centre 1,17,"ENTER 'EG' to substitute coded 'E' for clear 'G':"
1950 centre 1,18,"ENTER 'E<space>' to cancel a substitution:"
1960 centre 1,19,"or ENTER '99' to see original message."
1970 END DEFine instns
1980 REMark *****
1990 DEFine PROCedure dialog
2000 REPEAT dial
2010 b2:INK#0,4:centre 0,2,"":INPUT#0,di$:b2
2020 IF di$ = "99":RETURN
2030 IF LEN(di$) <> 2:b5:CLS#0,3:NEXT dial
2040 IF NOT di$(1) INSTR a$:b5:CLS#0,3:NEXT dial
2050 IF NOT di$(2) INSTR " " & a$:b5:CLS#0,3:NEXT dial
2060 EXIT dial
2070 END REPEAT dial
2080 INK 4
2090 flag=0
2100 FOR n=1 TO LEN(enc$)
2110 IF enc$(n)=di$(1):tri$(n)=di$(2):flag=1
2120 END FOR n
2130 IF NOT flag

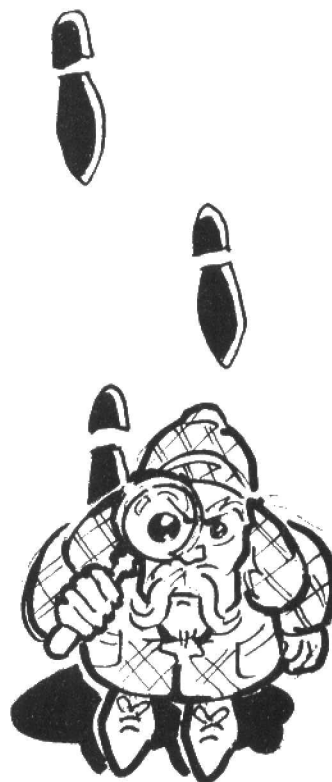
```



```

2140 INK#0,2
2150 centre 0,1,"No " & di$(1) & " in coded message- press a key"
2160 b5:PAUSE
2170 CLS#0
2180 dialog
2190 END IF
2200 AT 0,0:PRINT tri$:CLS#0
2210 dialog
2220 END DEFine dialog
2230 REMark *****
2240 DEFine PROCedure mon_window
2250 MODE 4:OPEN#1,con_512x256a0x0
2260 PAPER#1,0:INK#1,7:CLS#1
2270 CSIZE 0,0
2280 END DEFine mon_window
2290 REMark 85 chars per line
2300 REMark *****
2310 DEFine PROCedure tv_window
2320 WINDOW#0,448,40,32,216
2330 PAPER#0,0:INK#0,7
2340 OPEN#1,con_448x200a32x16_128
2350 PAPER#1,0:INK#1,7
2360 OPEN#2,con_448x200a32x16_128
2370 PAPER#2,1:INK#2,7
2380 MODE 4:CSIZE 0,0
2390 REMark 74 chars per line
2400 END DEFine tv_window
2410 REMark *****

```



TF Services

ASTRACOM Modem

Intelligent buffered modem with text status messages. Hayes protocol, parallel printer port (can be used as 6k serial to parallel printer buffer), 240vac or 9-12vac. Autodial/answer and many programmable registers incl. printer logging of RX and/or TX data. BT approved. Can be used with any computer.

NEW! V22/22BIS WITH V2.03 FIRMWARE! NEW!
 AC18000 (V21/22 v1.5a) £175
 AC18001 (V21/22/22 v2.03 + tone ding) ... £275
 AC18000 (V21/22/22/22bis V2.03) (tone) .. £299
 V22/V22bis upgrades ring for prices

Qualsoft Terminal

Viewdata/VT52-VT100 QL TERMINAL EMULATOR

Een deluxe communicatieprogramma van Qualsoft

Multitasking for electronic mail, PRESTEL etc, CET download, Phone directories for ALL (yes ALL) modems for the QL, autodial (where poss) and logon, two-way FILE TRANSFER to ATARI ST, IBM PC, Psion Organiser (via comes link), XMODEM, real time clock/timer, buffered logs to file/printer, transmit files, editable command line, Swedish/Hungarian/German options, translates, editor etc.

SOLVES ALL PACKAGED MODEM SOFTWARE PROBLEMS
 Unbuffered modems usable with miracle Modaptor

Software (3.5" or mdv) & manual £30

COMPUTER CLEANER

Much improved mains filters, 40-80 db cut & 3 spike filters, 390 joule cut (3 & 4 way)
 Certified K/Y 240ac capacitors

2-way (5a adaptor) £14
 3-way (5a adaptor) £18
 4-way (13a trailing socket) £24

All prices include VAT and postage in the UK

QL REPAIRS

QLs tested with Thorn EMI test rig and ROM software
 (MDV hardware extra if required)

£25 - 6 month guarantee

London STD codes

London 01- dialling code is changing to 071- or 081- in May 1990. Use our look-up tables and text (export file) conversion program to save HOURS of work.

Disks contain 5 text lists in database import formats (quote delimited, etc) & a program for text (ie export) file conversion. Also provided:

IBM PC : Files for major databases/spreadsheets
 QL : Psion Archive .dsk file

3.5/5.25 disk on microdrive - £12

QL SPARES

Faulty QL boards (excl plug in ICs) £9
 Keyboard membrane £7
 68008 cpu £8
 1377 PAL £25
 J11 row set £12
 JS row set £25
 8382 ULR £10
 8381 ULR £10
 8849 £7.50
 MDV ULR £12
 Other components (sockets etc) ... pls phone

FILE TRANSFER

Programs to transfer text AND programs error free (XMODEM) between computers

Available for IBM PC and compatibles, ATARI ST and Sinclair QL. Connects to Psion organiser via Psion comes link

Qualsoft program (per m/c) £7.50
 Serial lead (name 2 computers) £10
 Complete package for 2 computers, £25

MAIL ORDER ONLY

12 Bouverie Place, London W2 1RB (tel: 071-724 9053)

SCROLLING BUS FOR ORDERING-QL HELP - 071-706 2379 (all speeds to V22bis)

Fax: 071-706 2379 Telex: 265451 (ref: 72:W090645) Prestel: 017249053



£1000 REWARD

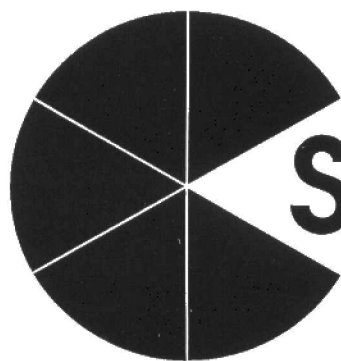


ANY INFORMATION ON PIRACY SHOULD BE PASSED TO F.A.S.T. (THE FEDERATION AGAINST SOFTWARE THEFT)

TELEPHONE 071-497 8973

FOR INFORMATION LEADING TO A PROSECUTION & CONVICTION





**BRAND NEW
MICRODRIVE CARTRIDGES
£2.00 EACH ANY QUANTITY**

Sector Software

The best programs and peripherals for the QL

OZ/QL to Z88 File Transfer

Software and cable to connect the Z88 and QL and transfer any files between them. Includes Archive to Pipedream and back conversion routines. **£25**

Amiga to Z88 File Transfer

Software and cable to connect the Z88 and Amiga and transfer any files between them **£25**

Spellbound

A spelling checker that checks your spelling AS YOU TYPE. Based on a 30,000 word dictionary, works with Quill or The Editor V1.17 onwards on the expanded QL. **£30**

Taskmaster

A brilliant multitasking front end system which lets you use the QL as a serious machine. Multitask many programs at once. **£25**

Files 2

File handling utility with scores of features. Written by Peter Jeffries. Ideal enhancement for Taskmaster users. **£12**

Write Turn

Turn spreadsheets and documents on their sides with this excellent utility, works on Epson and compatible printers. **£12**

QL World Index

A complete index to the contents of QL World from its start to May 1988. Find articles and reviews in seconds, 160K+ of data compressed to fit into a 128K QL. **£6**

Flashback

A very fast and slick database which has very few limitations. Will also convert Archive files. **£25**
Flashback Special Edition is a greatly advanced version with lots of extra features including report generator, mail merge, label printing, etc. **£40**

Touch Typist

Excellent typing tutor that works. 200 lessons, graph of your progress, adjustable difficulty levels. **£12**

Ferret

Find lost files fast with this file search utility which will read all your files on disk or mdv looking for a match with your search text. **£12**

STD Index

This index to all the dialling codes in the country executes from disk in 15 seconds. Know the place and it will tell you the number, know the number and it will tell you the place!
(Expanded QL only.) **£12**

Page Designer 2

This is a full feature desktop publisher that has to be seen to be believed. Ask for full details of this system and its support programs. **£35**

Phillips CM8833 Colour Stereo Monitor

A stereo monitor for the QL, Amiga, ST or almost any computer. **£260**

Cambridge Satellite TV System 2

48 channel stereo infra red remote control receiver with 60cm square dish. Receive all the current and planned future Astra transmissions. Now available for immediate delivery. Contact us for a leaflet on this system or call in and see the quality for yourselves.

SYSTEM II 1.8 DB LNB..... **£299**
GOLD SYSTEM 1.5 DB LNB..... **£349**

DRAMATIC PRICE REDUCTIONS ON STAR PRINTERS!

Star LC10 printer **£190**
Star LC10 colour printer **£230**
Star LC2410 (24 pin) printer **£274**
(Please add £6 if carrier delivery required)
Z88 Computer..... **£230**
Z88 All in one pack (Z88 128K RAM, carry case, power supply, batteries, user manual) **£287**

QL Keyboard membrane..... **£7**
QL Service Manual..... **£25**
ZX8301 ULA..... **£10**
3.5in. DSDD disks (each)..... **£1.25**
Cartridge Doctor (rescue corrupt MDV's)..... **£16**

New Price List available - Send for your free copy



Sector Software



Unit 13, Centurion Way Industrial Estate, Farington, Leyland, Lancs. PR5 2GU
Tel: (0772) 454328/452414 (2 lines), Fax: (0772) 454680